MDPI

Applied Metaheuristic Computing

Edited by
Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia,
Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Printed Edition of the Topics Published in *Applied Sciences, Symmetry, IJFS*

www.mdpi.com/topics

Applied Metaheuristic Computing

Applied Metaheuristic Computing

Editors

Peng-Yeng Yin
Ray-I Chang
Youcef Gheraibia
Ming-Chin Chuang
Hua-Yi Lin
Jen-Chun Lee

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade • Manchester • Tokyo • Cluj • Tianjin

Editors

Peng-Yeng Yin
Information Technology and
Management Program,
Ming Chuan University
Taiwan

Ray-I Chang
National Taiwan University
Taiwan

Youcef Gheraibia
University of York
United Kingdom

Ming-Chin Chuang
China University of
Technology
Taiwan

Hua-Yi Lin
China University of
Technology
Taiwan

Jen-Chun Lee
National Kaohsiung
University of Science and
Technology
Taiwan

Editorial Office
MDPI
St. Alban-Anlage 66
4052 Basel, Switzerland

This is a reprint of articles from the Topic published online in the open access journals *Applied Sciences* (ISSN 2076-3417), *Administrative Sciences* (ISSN 2076-3387), *Behavioral Sciences* (ISSN 2076-328X), and *Clean Technologies* (ISSN 2571-8797) (available at: https://www.mdpi.com/topics/Applied_Metaheuristic_Computing).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

LastName, A.A.; LastName, B.B.; LastName, C.C. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-0365-5569-0 (Hbk)
ISBN 978-3-0365-5570-6 (PDF)

Contents

applied sciences

MDPI

Editorial

Special Features and Applications on Applied Metaheuristic Computing

Peng-Yeng Yin [1],* and Ray-I Chang [2]

[1] Information Technology and Management Program, Ming Chuan University, No. 5 De Ming Rd., Gui Shan District, Taoyuan City 333, Taiwan
[2] Department of Engineering Science and Ocean Engineering, National Taiwan University, Taipei 106, Taiwan
* Correspondence: pengyengyin@gmail.com

Citation: Yin, P.-Y.; Chang, R.-I. Special Features and Applications on Applied Metaheuristic Computing. *Appl. Sci.* **2022**, *12*, 9342. https://doi.org/10.3390/app12189342

Received: 14 September 2022
Accepted: 14 September 2022
Published: 18 September 2022

1. Introduction

In recent years, many important yet complex problems, either continuous or combinatorial, suffer the intractability of the problem of nature. This is because the classic exact methods are constrained with strict assumptions, such as differentiability or linearity for continuous objective functions or constrained size for combinatorial problems. Instead of using exact methods such as calculus or mathematical programming, heuristics have been used as alternatives for seeking approximation solutions. However, it is argued that the error bound of the solutions obtained by heuristics to the optimal ones is usually loose and it is not acceptable for some accuracy-critical applications, as in finance and security domains. Moreover, the heuristics are problem-dependent and lack generalization for solving various genuine problems. Applied Metaheuristic Computing (AMC) has emerged as a prevailing optimization technique for tackling perplexing engineering and business problems. This is partly due to AMC's ability to guide the search course beyond the local optimality, which impairs the capability of traditional heuristics, and partly due to AMC providing general frameworks which can be applied to solve a broad range of problems. We have witnessed many successful AMC applications in various domains, such as scheduling, routing, ordering, bin packing, assignment, facility layout planning, renewable energy, portfolio optimization, classification, and forecasting, among others. The aim of this topic series was to collect quality papers proposing cutting-edge methodology and innovative applications which drive the advances of AMC. The manuscript submission to this topic series was closed on 31 March 2022. We received 116 submissions and 33 papers were published. The acceptance rate is 28%. Due to the success of this topic series, we are launching the second volume to collect more cutting-edge papers in the field of AMC.

2. Special Features on AMC

The most commonly seen AMC algorithms include genetic algorithm (GA), genetic programming (GP), evolutionary strategy (ES), evolutionary programming (EP), memetic algorithm (MA), particle swarm optimization (PSO), ant colony optimization (ACO), artificial bee colony (ABC), differential evolution (DE), firefly algorithm (FA), simulated annealing (SA), tabu search (TS), scatter search (SS), variable neighborhood search (VNS), and GRASP, to name a few. In this topic series, we considered special features which have been adopted to enhance the effectiveness of AMC. One notable feature is to apply metaheuristics to learn the architecture of neural networks or to optimize the hyperparameters of machine learning algorithms. The first paper, authored by J. Chou, T. Pham, and C. Ho, developed a metaheuristic-optimized machine-learning algorithm for multi-level classification in civil and construction engineering [1]. In particular, the FA fine-tunes the hyperparameters of the least squares support vector machine (LSSVM) to construct an optimized LSSVM multi-level classification model. The second paper, authored by R. Caraka, H. Yasin, R. Chen, N. Goldameir, B. Supatmanto, T. Toharudin, M. Basyuni, P. Gio, and B. Pardamean, used

GA to evolve a hybrid cascade neural network for space–time pollution data forecasting [2]. The third paper, authored by S. Jiao, C. Wang, R. Gao, Y. Li, and Q. Zhang, improved the Harris Hawks optimization algorithm by a multi-strategy search. The proposed method was used to optimize the LSSVM to model the reactive power output [3].

The other fast-growing research direction in AMC is developing a multi-objective optimization framework for various metaheuristics. The paper authored by I. Masich, M. Kulachenko, P. Stanimirović, A. Popov, E. Tovbis, A. Stupina, and I. Kazakovtsev proposed a multi-criteria genetic algorithm for pattern generation in logical data analysis. The proposed method has a flexibility of allowing the pattern to maximally cover the objects in the target class and minimize the covered objects in the opposite class. The generated patterns by using the multi-criteria genetic algorithm are more meaningful than using the traditional fuzzy approaches [4]. The paper by A. Alqaili, M. Qais, and A. Al-Mansour developed a new discrete multi-objective integer search algorithm to optimize the road pavement performance and minimize the maintenance cost at the same time [5]. The paper by B. Changaival, K. Lavangnananda, G. Danoy, D. Kliazovich, F. Guinand, M. Brust, J. Musial, and P. Bouvry applied the NSGA-II, which is a famous multi-objective version of GA, for determining the optimal station locations of carsharing fleet service [6]. The objectives for setting fleet stations include maximum user coverage, least walking distance, and flexibility for returning. The paper authored by R. Díaz, E. Solares, V. de-León-Gómez, and F. Salas tackled the portfolio optimization problem by using a three-phase multi-objective approach [7]. In the first phase, an artificial neural network was constructed to predict the stock price. Then, an EA was applied to select the stocks. Finally, MOEA/D, which is a multi-objective EA, was adopted to optimize the stock portfolio considering multiple criteria.

3. Special Applications on AMC

In addition to classic AMC applications, we saw some special applications in this topic series. There are four papers which address efficient methods of energy management. The first paper by H. Lin, P. Wang, W. Lin, K. Chao, and Z. Yang analyzed the attack sources of robot networks (botnets) for a renewable energy management system [8]. The authors applied a revised locust swarm optimization algorithm to search near-global optima of the most probable attack paths via the internet protocol traceback schemes. The second paper, authored by M. Kara, A. Laouid, M. AlShaikh, M. Hammoudeh, A. Bounceur, R. Euler, A. Amamra, and B. Laouid proposed a multi-round Proof of Work (PoW) consensus algorithm for preserving energy consumption and resisting attacks [9]. The other two papers deal with power microgrid operations. The paper authored by T. Nguyen, T. Ngo, T. and Dao, T. Nguyen proposed an improved sparrow search algorithm by incorporating the mutation mechanism of the firefly algorithm [10]. The improved version ensures the share of green power generation and a safe symmetry power grid among distributed clean power sources. The last paper, authored by N. Ning, Y. Liu, H. Yang, and L. Li presented a scheme for efficiently running an energy storage station. An improved aquila optimizer for the optimal configuration of the combined cooling, heating, and power microgrid was proposed to symmetrically enhance the economic and environmental protection performance [11].

AMC has intensively contributed to improve information security. The paper by R. Abu Khurma, I. Almomani, and I. Aljarah, proposed an IoT botnet intrusion detection system (IDS) by hybridizing the salp swarm algorithm (SSA) and ant lion optimization (ALO) [12]. The proposed method outperformed several existing approaches in terms of standard performance measures. The paper authored by L. Wang, L. Gu, and Y. Tang developed an IDS to deal with massive redundant alarms when monitoring the frequent occurrence of network security events [13]. The proposed method uses the whale optimization algorithm to conduct an alarm hierarchical clustering. The results showed that the proposed algorithm can effectively reduce the load of IDS and staff. The paper authored by A. Aldallal, and F. Alisa presented another IDS to secure data in cloud cyberspace using a hybrid metaheuristic. The method combines a support vector machine and a GA to enhance the detection rate [14]. The paper by M. Madbouly, Y. Mokhtar, and S. Darwish employed a quantum game theory approach to select the optimal recovery method for mobile databases as a response to environmental failures in mobile computing such as the number of processes, the time needed to send messages, and the number of messages logged-in time [15]. The paper authored by J. Pan, X. Sun, H. Yang, V. Snášel, and S. Chu introduced a two-level mechanism and look-up table approach to solve the problem of sufficient diversity of features for information hiding [16]. The method has better capacity and image quality such that the storage and security are ensured. The paper by M. Ali, A. Ismail, H. Elgohary, S. Darwish, and S. Mesbah employed fuzzy hash within blockchain for tracing Chain of Custody (CoC) in preserving digital evidence in cybercrime [17]. The fuzzy hash functions within blockchain can demonstrate the CoC evidence has not been tampered. Software reliability is also an important issue in information security. The paper authored by Y. Kim, K. Song, H. Pham, and I. Chang developed a software reliability model where the software failures are interdependent and a software failure, if not immediately detected, can cause a sequence of failures and resulting massive losses [18]. The paper authored by E. Sakalauskas, I. Timofejeva, and A. Kilciauskas presented a new sigma-identification protocol based on matrix power function (MPF). The authors showed that the given protocol ensures the NP-completeness of MPF and it is resistant against direct and eavesdropping attacks [19].

Two papers were devoted to determining the optimal assembly sequence of mechanical parts. The first paper, authored by M. Suszyński and K. Peta, determined the best assembly sequence model by clustering 10,000 artificial neural networks which were created by using various network-training methods and activation functions [20]. The proposed model can predict the optimal assembly time of mechanical parts. The second paper, authored by M. Suszyński, K. Peta, V. Černohlávek, and M. Svoboda, on the other hand, creates a large number of artificial neural networks with a different means. It considers various criteria of Design for Assembly (DFA) as the input data and then predicts the assembly time [21].

Object detection and recognition has long been an interesting area for classic AMC applications. The paper by C. Ticala, C. Pintea, and O. Matei provided a new edge detection method for medical images [22]. The method is based on a sensitive ACO where a vector called pheromone sensitivity level is used to control the ant's sensibility to the pheromone attraction. The paper authored by D. Wang, J. Ni, and T. Du presented an image recognition method for coal gangue recognition [23]. The adaptive shrinking grid search chaos wolf optimization algorithm was proposed to optimize the parameters of the neural network to enhance the image recognition accuracy. The paper authored by A. Alzbier and C. Chen constructed a denoising network consisting of a kernel prediction network and a deep generative adversarial network. Compared with the state-of-the-art result, the proposed denoising network has a better denoising effect and shorter running time [24]. The paper by V. Hrytsyk, M. Medykovskyy, and M. Nazarkevych provided a subjective assessment of various edge-detection filters for reproducing the object image in a room under different lighting conditions during educational activity in Ukraine [25]. It was found that there exist dependencies between priority to certain filters and the lighting condition.

The rest of the papers in this topic series address miscellaneous applications. The paper authored by E. Matos, R. Parmezan Bonidia, D. Sipoli Sanches, R. Santos Pozza, and L. Dias Hiera Sampaio compared the performance of several heuristics including GA, PSO, and VNS for increasing the throughput of sequence allocation schemes in massive multi-user MIMO 5G networks [26]. The paper authored by L. Lara-Valencia, D. Caicedo, and Y. Valencia-Gonzalez presented a whale optimization algorithm (WOA) for the design of tuned mass dampers under earthquake excitations [27]. Multiple objectives including reducing the maximum horizontal displacement and the root mean square of displacements of the structures were implemented to improve the seismic safety. The paper by G. Csányi, D. Nagy, R. Vági, J. Vadász, and T. Orosz addressed the challenging issues and symmetrical problems of legal document anonymization. The existing methods are reviewed and illustrated by case studies from the Hungarian legal practice [28]. The paper authored by S. Liu, Z. Zhou, S. Dai, I. Iqbal, and Y. Yang presented a fast computation method for the green function which represents the seismic fields [29]. The computation method was able to evaluate the Sommerfeld integral efficiently and accurately and the result showed that the computational time was reduced by about 40%. The paper authored by X. Shen and D. Ihenacho developed a hybrid metaheuristic algorithm by combining DE and PSO for resolving the complex mathematical models of gas cyclone design [30]. The paper by M. Li, H. Sang, P. Liu, and G. Huang provided new practical criteria for identifying the positive definiteness of H-tensors [31]. An application and several numerical examples were provided to illustrate the effectiveness of the method. The paper by Y. Alotaibi presents a new clustering algorithm based on tabu search (TS) and adaptive search memory (ASM) [32]. As k-means clustering result is easily spoiled by the biased initial seeds, the TS and ASM were applied to search the elite initial seeds and irritate k-means clustering from there. Finally, the paper by Z. Lian, D. Luo, B. Dai, and Y. Chen presented a new discrete search algorithm based on Levy random distribution in order to obtain an escape from the local optima. The simulation results showed the fast convergence and search effectiveness of the proposed scheme [33].

Author Contributions: Conceptualization, P.-Y.Y. and R.-I.C.; writing—original draft preparation, P.-Y.Y. and R.-I.C.; writing—review and editing, P.-Y.Y. and R.-I.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: I would like to thank all the contributing authors for their innovative and well managed works. This topic series would not be possible without their enthusiasm for charting advanced AMC research. My thanks are also given to the anonymous reviewers for their unselfish and professional reviews which have helped shape the standard of this topic series. Finally, I am grateful to all the guest editors and the editorial team of MDPI.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chou, J.; Pham, T.; Ho, C. Metaheuristic Optimized Multi-Level Classification Learning System for Engineering Management. *Appl. Sci.* **2021**, *11*, 5533. [CrossRef]
2. Caraka, R.; Yasin, H.; Chen, R.; Goldameir, N.; Supatmanto, B.; Toharudin, T.; Basyuni, M.; Gio, P.; Pardamean, B. Evolving Hybrid Cascade Neural Network Genetic Algorithm Space–Time Forecasting. *Symmetry* **2021**, *13*, 1158. [CrossRef]
3. Jiao, S.; Wang, C.; Gao, R.; Li, Y.; Zhang, Q. Harris Hawks Optimization with Multi-Strategy Search and Application. *Symmetry* **2021**, *13*, 2364. [CrossRef]
4. Masich, I.; Kulachenko, M.; Stanimirović, P.; Popov, A.; Tovbis, E.; Stupina, A.; Kazakovtsev, L. Formation of Fuzzy Patterns in Logical Analysis of Data Using a Multi-Criteria Genetic Algorithm. *Symmetry* **2022**, *14*, 600. [CrossRef]
5. Alqaili, A.; Qais, M.; Al-Mansour, A. Integer Search Algorithm: A New Discrete Multi-Objective Algorithm for Pavement Maintenance Management Optimization. *Appl. Sci.* **2021**, *11*, 7170. [CrossRef]
6. Changaival, B.; Lavangnananda, K.; Danoy, G.; Kliazovich, D.; Guinand, F.; Brust, M.; Musial, J.; Bouvry, P. Optimization of Carsharing Fleet Placement in Round-Trip Carsharing Service. *Appl. Sci.* **2021**, *11*, 11393. [CrossRef]
7. Díaz, R.; Solares, E.; de-León-Gómez, V.; Salas, F. Stock Portfolio Management in the Presence of Downtrends Using Computational Intelligence. *Appl. Sci.* **2022**, *12*, 4067. [CrossRef]
8. Lin, H.; Wang, P.; Lin, W.; Chao, K.; Yang, Z. Identifying the Attack Sources of Botnets for a Renewable Energy Management System by Using a Revised Locust Swarm Optimisation Scheme. *Symmetry* **2021**, *13*, 1295. [CrossRef]
9. Kara, M.; Laouid, A.; AlShaikh, M.; Hammoudeh, M.; Bounceur, A.; Euler, R.; Amamra, A.; Laouid, B. A Compute and Wait in PoW (CW-PoW) Consensus Algorithm for Preserving Energy Consumption. *Appl. Sci.* **2021**, *11*, 6750. [CrossRef]
10. Nguyen, T.; Ngo, T.; Dao, T.; Nguyen, T. Microgrid Operations Planning Based on Improving the Flying Sparrow Search Algorithm. *Symmetry* **2022**, *14*, 168. [CrossRef]
11. Ning, N.; Liu, Y.; Yang, H.; Li, L. Design and Optimization of Combined Cooling, Heating, and Power Microgrid with Energy Storage Station Service. *Symmetry* **2022**, *14*, 791. [CrossRef]
12. Abu Khurma, R.; Almomani, I.; Aljarah, I. IoT Botnet Detection Using Salp Swarm and Ant Lion Hybrid Optimization Model. *Symmetry* **2021**, *13*, 1377. [CrossRef]
13. Wang, L.; Gu, L.; Tang, Y. Research on Alarm Reduction of Intrusion Detection System Based on Clustering and Whale Optimization Algorithm. *Appl. Sci.* **2021**, *11*, 11200. [CrossRef]

14. Aldallal, A.; Alisa, F. Effective Intrusion Detection System to Secure Data in Cloud Using Machine Learning. *Symmetry* **2021**, *13*, 2306. [CrossRef]
15. Madbouly, M.; Mokhtar, Y.; Darwish, S. Quantum Game Application to Recovery Problem in Mobile Database. *Symmetry* **2021**, *13*, 1984. [CrossRef]
16. Pan, J.; Sun, X.; Yang, H.; Snášel, V.; Chu, S. Information Hiding Based on Two-Level Mechanism and Look-Up Table Approach. *Symmetry* **2022**, *14*, 315. [CrossRef]
17. Ali, M.; Ismail, A.; Elgohary, H.; Darwish, S.; Mesbah, S. A Procedure for Tracing Chain of Custody in Digital Image Forensics: A Paradigm Based on Grey Hash and Blockchain. *Symmetry* **2022**, *14*, 334. [CrossRef]
18. Kim, Y.; Song, K.; Pham, H.; Chang, I. A Software Reliability Model with Dependent Failure and Optimal Release Time. *Symmetry* **2022**, *14*, 343. [CrossRef]
19. Sakalauskas, E.; Timofejeva, I.; Kilciauskas, A. Sigma Identification Protocol Construction Based on MPF. *Symmetry* **2021**, *13*, 1683. [CrossRef]
20. Suszyński, M.; Peta, K. Assembly Sequence Planning Using Artificial Neural Networks for Mechanical Parts Based on Selected Criteria. *Appl. Sci.* **2021**, *11*, 10414. [CrossRef]
21. Suszyński, M.; Peta, K.; Černohlávek, V.; Svoboda, M. Mechanical Assembly Sequence Determination Using Artificial Neural Networks Based on Selected DFA Rating Factors. *Symmetry* **2022**, *14*, 1013. [CrossRef]
22. Ticala, C.; Pintea, C.; Matei, O. Sensitive Ant Algorithm for Edge Detection in Medical Images. *Appl. Sci.* **2021**, *11*, 11303. [CrossRef]
23. Wang, D.; Ni, J.; Du, T. An Image Recognition Method for Coal Gangue Based on ASGS-CWOA and BP Neural Network. *Symmetry* **2022**, *14*, 880. [CrossRef]
24. Alzbier, A.; Chen, C. DGAN-KPN: Deep Generative Adversarial Network and Kernel Prediction Network for Denoising MC Renderings. *Symmetry* **2022**, *14*, 395. [CrossRef]
25. Hrytsyk, V.; Medykovskyy, M.; Nazarkevych, M. Estimation of Symmetry in the Recognition System with Adaptive Application of Filters. *Symmetry* **2022**, *14*, 903. [CrossRef]
26. Matos, E.; Parmezan Bonidia, R.; Sipoli Sanches, D.; Santos Pozza, R.; Dias Hiera Sampaio, L. Pilot Sequence Allocation Schemes in Massive MIMO Systems Using Heuristic Approaches. *Appl. Sci.* **2022**, *12*, 5117. [CrossRef]
27. Lara-Valencia, L.; Caicedo, D.; Valencia-Gonzalez, Y. A Novel Whale Optimization Algorithm for the Design of Tuned Mass Dampers under Earthquake Excitations. *Appl. Sci.* **2021**, *11*, 6172. [CrossRef]
28. Csányi, G.; Nagy, D.; Vági, R.; Vadász, J.; Orosz, T. Challenges and Open Problems of Legal Document Anonymization. *Symmetry* **2021**, *13*, 1490. [CrossRef]
29. Liu, S.; Zhou, Z.; Dai, S.; Iqbal, I.; Yang, Y. Fast Computation of Green Function for Layered Seismic Field via Discrete Complex Image Method and Double Exponential Rules. *Symmetry* **2021**, *13*, 1969. [CrossRef]
30. Shen, X.; Ihenacho, D. Design of Gas Cyclone Using Hybrid Particle Swarm Optimization Algorithm. *Appl. Sci.* **2021**, *11*, 9772. [CrossRef]
31. Li, M.; Sang, H.; Liu, P.; Huang, G. Practical Criteria for H-Tensors and Their Application. *Symmetry* **2022**, *14*, 155. [CrossRef]
32. Alotaibi, Y. A New Meta-Heuristics Data Clustering Algorithm Based on Tabu Search and Adaptive Search Memory. *Symmetry* **2022**, *14*, 623. [CrossRef]
33. Lian, Z.; Luo, D.; Dai, B.; Chen, Y. A Levy Distribution Based Searching Scheme for the Discrete Targets in Vast Region. *Symmetry* **2022**, *14*, 272. [CrossRef]

applied sciences

MDPI

Article

Metaheuristic Optimized Multi-Level Classification Learning System for Engineering Management

Jui-Sheng Chou [1,*], Trang Thi Phuong Pham [1,2] and Chia-Chun Ho [1]

[1] Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei 106335, Taiwan; M10405811@mail.ntust.edu.tw (T.T.P.P.); cchocv@mail.ntust.edu.tw (C.-C.H.)
[2] Department of Civil Engineering, University of Technology and Education—The University of Da Nang, Da Nang 550000, Vietnam
* Correspondence: jschou@mail.ntust.edu.tw

Abstract: Multi-class classification is one of the major challenges in machine learning and an ongoing research issue. Classification algorithms are generally binary, but they must be extended to multi-class problems for real-world application. Multi-class classification is more complex than binary classification. In binary classification, only the decision boundaries of one class are to be known, whereas in multiclass classification, several boundaries are involved. The objective of this investigation is to propose a metaheuristic, optimized, multi-level classification learning system for forecasting in civil and construction engineering. The proposed system integrates the firefly algorithm (FA), metaheuristic intelligence, decomposition approaches, the one-against-one (OAO) method, and the least squares support vector machine (LSSVM). The enhanced FA automatically fine-tunes the hyperparameters of the LSSVM to construct an optimized LSSVM classification model. Ten benchmark functions are used to evaluate the performance of the enhanced optimization algorithm. Two binary-class datasets related to geotechnical engineering, concerning seismic bumps and soil liquefaction, are then used to clarify the application of the proposed system to binary problems. Further, this investigation uses multi-class cases in civil engineering and construction management to verify the effectiveness of the model in the diagnosis of faults in steel plates, quality of water in a reservoir, and determining urban land cover. The results reveal that the system predicts faults in steel plates with an accuracy of 91.085%, the quality of water in a reservoir with an accuracy of 93.650%, and urban land cover with an accuracy of 87.274%. To demonstrate the effectiveness of the proposed system, its predictive accuracy is compared with that of a non-optimized baseline model, single multi-class classification algorithms (sequential minimal optimization (SMO), the Multiclass Classifier, the Naïve Bayes, the library support vector machine (LibSVM) and logistic regression) and prior studies. The analytical results show that the proposed system is promising project analytics software to help decision makers solve multi-level classification problems in engineering applications.

Keywords: machine learning; multi-level classification; metaheuristic optimization; swarm and evolutionary algorithm; chaotic maps and Lévy flights; hybrid computing system; engineering management; civil and construction engineering

Citation: Chou, J.-S.; Pham, T.T.P.; Ho, C.-C. Metaheuristic Optimized Multi-Level Classification Learning System for Engineering Management. *Appl. Sci.* **2021**, *11*, 5533. https://doi.org/10.3390/app11125533

Academic Editor: José Machado

Received: 24 April 2021
Accepted: 10 June 2021
Published: 15 June 2021

1. Introduction

A considerable amount of research in the field of machine learning (ML) is concerned with developing methods that automate classification tasks [1]. Classification tasks are involved in several real-world applications, in such fields as civil engineering [2,3], medicine [4], land use [5], energy [6], investment [7], and marketing [8]. It is obvious that problems in the engineering domain are multi-class issues. Hence, there is a need to establish a learning framework for solving multi-level classification problems efficiently and effectively, which is the primary purpose of this study.

Various classification approaches have been proposed and used to solve real-life problems, ranging from statistical methods to ML techniques, such as linear classification (Naive

Bayes classifier and logistic regression), distance estimation (k-nearest neighbors), support vector machines (SVM), rule and decision-tree-based methods, and neural networks, to name a few [9]. Some studies have used fuzzy synthetic evaluation to classify seismic damage and assess risks to mountain tunnels [10], while others have used artificial neural networks (ANNs), SVM, Bayesian networks (Bayes Net) and classification trees (C5.0) to classify information that bears on project disputes and possible resolutions [11].

Nevertheless, many studies have also demonstrated that machine learning methods cannot solve multi-level classification problems efficiently or do not yield suitable forecasts for practical applications [12–15]. For example, the k-nearest neighbors (KNN) method is a lazy learner and very slow; a decision tree (DT) is good for classification problems but becomes complex to interpret if the tree grows largely, leading to overfitting.

Multi-level or multi-class classification problems are typically more difficult to solve than binary-class problems because the decision boundary in a multi-class classification problem tends to be more complex than that in a binary classification problem [16]. Therefore, it is preferable to break down a multi-class problem into several two-class problems and combine the output of these binary classifiers to obtain the final, multi-class decision [17].

Decomposition strategies [13] are commonly used to solve classification problems with multiple classes. These methods transform a multi-class classification problem into several binary classification problems [16]. Thus, many machine learning methods were applied with decomposition strategies, such as one-against-rest [18] and one-against-one [19], to improve the results.

One-against-one (OAO) and one-against-rest (OAR) are the most widely used decomposition strategies. The literature [19–21] compares some OAO and OAR classifiers that are based on single classification algorithms, including ANN, DT, KNN, linear discriminant analysis (LDA), logistic regression (LR), and SVM, and indicates that single classification algorithms combined with the OAO approach usually outperform those combined with the OAR approach.

Studies of binary classification regard the SVM as one of the most effective machine learning algorithms for classification [22,23]. The SVM is an algorithm with the potential to support increasingly efficient methods for multi-class classification. In particular, the OAO strategy has been used with very well-known software tools to model multi-class problems for SVM. For the SVM, the OAO method generally outperforms the OAR and other SVM-based multi-class classification algorithms [16,24,25]. Therefore, integrating OAO with the SVM yields a method (OAO-SVM) that is potentially effective for solving multi-class classification problems.

However, one of main challenges for the classical SVM is its high computational complexity, because the algorithm itself involves constrained optimization programming. The least squares support vector machine (LSSVM) is a highly enhanced machine-learning technique with many advanced features that support generalization and fast computation [26]. Empirical studies have suggested that LSSVMs are at least as accurate as conventional SVMs but with higher computing efficiency [27].

To improve the predictive accuracy of the LSSVM model, the parameters of the LSSVM must be optimized because the performance of the LSSVM depends on the selected regularization parameter (C) and the kernel function parameter (σ), which are known as LSSVM hyperparameters. Modern evolutionary optimization (EA) techniques appear to be more efficient in solving constrained optimization problems because of their ability to seek the global optimal solution [28].

Researchers always seek to improve the effectiveness of the methods that they use. Metaheuristics have become a popular approach in tackling the complexity of practical optimization problems [29–33]. Owing to the continuous development of artificial intelligence (AI) technology, many intelligent algorithms are now used in parameter optimization, including the genetic algorithm (GA) [34] and the particle swarm optimization algorithm (PSO) [35]. Many studies have also shown that the firefly algorithm (FA) can solve optimization problems more efficiently than can conventional algorithms, including GA and PSO [36,37].

In this study, metaheuristic components are incorporated into the standard FA to improve its ability to find the optimal solution. The efficiency of the optimized method (i.e., enhanced FA) was verified using many classic benchmark functions. Therefore, a new hybrid classification model (Optimized-OAO-LSSVM) that combines the OAO algorithm for decomposition and the enhanced FA to optimize the hyperparameters for solving multi-class engineering problems is established.

To validate the accuracy of prediction of the proposed Optimized-OAO-LSSVM model, its prediction performance was compared with that of previously proposed methods and other multi-class classification models. After the optimized classification model is verified, an intelligent and user-friendly system that can classify multi-class data in the fields of civil and construction engineering is developed.

The rest of this study is organized as follows. Section 2 introduces the context of this investigation by reviewing the relevant literature. Section 3 then describes all methods that are used to develop the proposed system and to establish its effectiveness. Section 4 elucidates the metaheuristic optimized multi-level classification system. Section 5 validates the system using case studies in the areas of civil engineering and construction management. Section 6 draws conclusions and presents the contributions of this study.

2. Literature Review

Data mining (DM) is the process of analyzing data from various perspectives and extracted useful information. DM involves methods at the intersection of AI, ML, statistics, and database systems. To extract information and the characteristics of data from databases, almost all DM research focuses on developing AI or ML algorithms that improve the computing time and accuracy of prediction models [38,39].

AI-based methods are strong, efficient tools for solving real-world engineering problems. Many AI techniques are applied in construction engineering and construction management [40,41] and they are usually used to handle prediction and classification problems. For example, ANN was combined with PSO to create a new model in the prediction of laser metal deposition process [42]. Moreover, to enhance the water quality predictions, Noori et al. [43] developed a hybrid model by combining a process-based watershed model and ANN. In terms of structural failure, Mangalathu et al. [44] contributed to the critical need of failure mode prediction for circular reinforced concrete bridge columns by using several AI algorithms, including nearest neighbors, decision trees, random forests, Naïve Bayes, and ANN.

SVM is one of powerful AI techniques in solving pattern recognition problems [45]. For instance, SVM-based classification model is used to forecast soil quality [46], relevance vector regression (RVR) and the SVM is used to predict the rock mass rating of tunnel host rocks [47]. Biomonitoring and the multiclass SVM are used to evaluate the quality of water [48]. Additionally, Du et al. [49] combined the dual-tree complex wavelet transform (DT-CWT) and modified matching pursuit optimization with an multiclass SVM ensemble (MPO-SVME) to classify engineering surfaces.

In this work, OAO was used for decomposition [21]. This method is even effective to handle a multi-class classification problem because it involves solving several binary sub-problems that are easier to solve than the original problem [16,50]. Many combined mechanisms for implementing the OAO strategy exist; they include the voting OAO (V-OAO) strategy and the weighted voting OAO (WV-OAO) strategy [16,21,51].

However, the most intuitive combination is a voting strategy in which each classifier votes for the predicted class and the class with the most votes is output by the system. In building binary classifiers for each approach, various methods can be used to combine with output of OAO to yield the ultimate solution to problems that involve multiple classes [16]. Zhou et al. [52] combined the OAO scheme with seven well-known binary classification methods to develop the best model for predicting the different risk levels of

Chinese companies. Galar et al. [20] used distance-based relative competence weighting and combination for OAO to solve multi-class classification problems.

Suykens et al. [53] improved the LSSVM and demonstrated that it solves nonlinear estimation problems. The LSSVM solves linear equations rather than the quadratic programming problem. Some studies have demonstrated the superiority of the LSSVM over the standard SVM [54,55]. In the present investigation, multi-class datasets are used to demonstrate that the LSSVM is more effective than the SVM when each is combined with the OAO strategy. Likewise, the main shortcoming of LSSVM is the need to set its hyperparameters. Hence, a means of automatically evaluating the hyperparameters of the LSSVM while ensuring its generalization performance is required. The hyperparameters of a model have a critical effect on its predictive accuracy. Favorably, metaheuristic algorithms constitute the most effective means of tuning hyperparameters.

The firefly algorithm (FA) [56] is shown to be effective for solving optimization problems. The FA has outperformed some metaheuristics, such as the genetic algorithm, particle swarm optimization, simulation annealing, ant colony optimization and bee colony algorithms [57,58]. Khadwilard et al. [59] presented the use of FA in parameter setting to solve the job shop scheduling problem (JSSP). They concluded that the FA with parameter tuning yielded better results than the FA without parameter tuning. Aungkulanon et al. [60] compared the performance metrics of the FA, such as processing time, convergence speed and quality of the results, with those of the PSO. The FA is consistently superior to PSO in terms of both ease of application and parameter tuning.

Hybrid algorithms are observed to outperform their counterparts in classification [4,61]. In the last decade, much work has been done in solving multi-class classification problems using hybrid algorithms [62,63]. Seera et al. [64] proposed a hybrid system that comprises the Fuzzy MinMax neural network, the classification and regression tree, and the random forest model for performing multiple classification. Tian et al. [65] combined the SVM with three optimizing algorithms—grid search (GS), GA and PSO—to classify faults in steel plates. Chou et al. [62] combined fuzzy logic (FL), a fast and messy genetic algorithm (fmGA), and SVMs to improve the classification accuracy of project dispute resolution.

Therefore, this study proposes a new hybrid model that integrates an enhanced FA into the LSSVM combined with the voting OAO scheme, called the Optimized-OAO-LSSVM, to solve multi-class classification problems.

3. Methodology

In this section, several methods are introduced to create a metaheuristic optimized multi-level classification system for predicting multi-class classification, involving a decomposition strategy, a hybrid model of metaheuristic optimization in machine learning, and performance measures.

3.1. Decomposition Methods

The strategy of decomposing the original problem into many sub-problems is extensively used in applying binary classifiers to solve multi-class classification problems. The OAO algorithm was used for decomposition herein. The OAO scheme divides an original problem into as many binary problems as possible pairs of classes. Each problem is faced by a binary classifier, which is responsible for distinguishing between each of the pair, and then the outputs of these base classifiers are combined to predict the final output.

Specifically, the OAO method constructs k(k–1)/2 classifiers [16], where k is the number of classes. Classifier *ij*, named *fij*, is trained, using all of the patterns from class *i* as positive instances. All of the patterns from class *j* are negative cases and the rest of the data points are ignored. The code-matrix in this case has dimensions k × k(k–1)/2 and each column corresponds to a binary classifier of a pair of classes. All classifiers are combined to yield the final output.

Different methods can be used to combine the obtained classifiers for the OAO scheme. The most common method is a simple voting method [66] by which a group, such as people in a meeting or an electorate, makes a decision or expresses an opinion, usually following discussion, debate or election campaigns.

3.2. Optimization in Machine Learning

3.2.1. Least Squares Support Vector Machine for Classification

The least squares SVM (LSSVM), proposed by Suykens et al. [53], is an enhanced ML technique with many advanced features. Therefore, the LSSVM has high generalizability and a low computational burden. In a function estimation of the LSSVM, given a training dataset $\{x_k, y_k\}_{k=1}^{N}$, the optimization problem is formulated as follows:

$$\min_{\omega,b,e} J(\omega, e) = \frac{1}{2}\|\omega\|^2 + \frac{1}{2}C\sum_{k=1}^{N} e_k^2 \tag{1}$$

subject to $y_k = \langle \omega, \varphi(x_k) \rangle + b + e_k,\ k = 1, \ldots N$
where *J(ω,e)* is the optimization function; ω is the parameter in the linear approximation; $e_k \in$ R are error variables; $C \geq 0$ is a regularization constant that represents the trade-off between the empirical error and the flatness of the function; x_k is the input patterns; y_k are prediction labels; and *N* is the sample size.

Equation (2) is the resulting LSSVM model for function estimation.

$$f(x) = \sum_{k=1}^{N} \alpha_k K(x, x_k) + b \tag{2}$$

where α_k, b are Lagrange multipliers and the bias term, respectively; and *K(x, x_k)* is the kernel function.

The Gaussian radial basis function (RBF) and the polynomial are commonly used kernel functions. RBFs are more frequently used because, unlike linear kernel functions, they can classify multi-dimensional data efficiently. Therefore, in this study, an RBF kernel is used. Equation (3) is the RBF kernel.

$$K(x, x_k) = \exp(-\|x - x_k\|^2 / 2\sigma^2 \tag{3}$$

Although the LSSVM can effectively learn patterns from data, the main shortcoming is that the predictive accuracy of an LSSVM model depends on the setting of its hyperparameters. Parameter optimization in an LSSVM includes the regularization parameter (*C*) in Equation (1) and the sigma of the RBF kernel (σ) in Equation (3). The generalizability of the LSSVM can be increased by determining optimal values of *C* and σ. In this investigation, the enhanced FA, which is an improved stochastic, nature-inspired metaheuristic algorithm, was developed to finetune the above hyperparameters *C* and σ.

3.2.2. Enhanced Firefly Algorithm

In this study, the enhanced firefly algorithm is proposed to improve the LSSVM's hyperparameters. The FA is improved by integrating stochastic agents to enrich global exploration and local exploitation.

Metaheuristic Firefly Algorithm

Yang (2008) developed the FA, which is inspired by the swarm nature of fireflies [67]. This algorithm is designed to solve global optimization problems in which each individual firefly in a population interacts with each other through their light intensity. The attractiveness of an individual firefly is proportional to its intensity. Visibly, the less this attraction for another individual firefly, the farther away it is from its location.

Despite the effectiveness of conventional FA in solving optimization problems, it often gets stuck in the local optima [39]. Randomization is considered an important part of searching optimal solutions. Therefore, fine-tuning the degree of randomness and balancing the local and global search are critical for the favorable performance of a metaheuristic algorithm.

The achievement of the FA is decided by three parameters, which are β, γ, and α, where β is the attractiveness of a firefly, γ is the absorption coefficient, and α is a trade-off constant to determine the random movements. Hence, this study supplements metaheuristic components—chaotic maps, adaptive inertia weight (AIW) and Lévy flight—into the basic FA. The components are not only to restore the balance between exploration and exploitation but also to increase the probability of escaping from the attraction of local optima.

Chaotic Maps: Generating a Variety of Initial Population and Refining Attractive Values

The simplest chaotic mapping operator is the logistic mapping, which creates more diversity than randomly selected baseline populations, and reduces the probability of early convergence [68]. The logistic map is formulated as Equation (4).

$$X_{n+1} = \eta X_n(1 - X_n) \tag{4}$$

where n is the number label of a firefly and X_n is the logistic chaotic value of the n^{th} firefly. In this work, initial populations are generated using the logistic map equation, and parameter η is set to 4.0 in all experiments.

Additionally, chaotic maps are used as efficient alternatives to pseudorandom sequences in chaotic systems [69]. A Gauss/mouse map is the best chaotic map for tuning the attractiveness parameter *(β)* of the original FA. Equation (5) describes the Gauss/mouse map that was used in this study.

$$\text{Gauss/mouse map}: \ \beta^t_{chaos} = \begin{cases} 0 & \beta^{t-1}_{chaos} = 0 \\ 1/\beta^{t-1}_{chaos} \bmod(1) & otherwise \end{cases} \tag{5}$$

The β of a firefly is updated using Equation (6).

$$\beta = (\beta^t_{chaos} - \beta_0)e^{-\gamma r_{ij}{}^2} + \beta_0 \tag{6}$$

where β is the firefly attractiveness; β^t_{chaos} is the t^{th} Gauss/mouse chaotic number and t is the iteration number; β_0 is the attractiveness of the firefly at distance $r = 0$; r_{ij} is the distance between the i^{th} firefly and the j^{th} firefly; e is a constant coefficient, and γ is the absorption coefficient.

Adaptive Inertia Weight: Controlling Global and Local Search Capabilities

In this investigation, the AIW was integrated into the original FA because AIW has critical effects on not only the optimal solution convergence, but also the computation time. A monotonically decreasing function of the inertia weight was used to change the randomization parameter α in the conventional FA. The AIW was utilized to adjust the parameter α by which the distances between fireflies were reduced to a reasonable range (Equation (7)).

$$\alpha^t = \alpha_0 \theta^t \tag{7}$$

where α_0 is the initial randomization parameter; α^t is the randomization parameter in the t^{th} generation; θ is the randomness reduction constant ($0 < \theta < 1$), and t is the number of the iteration. The selected value of θ in this implementation is 0.9 based on the literature, and $t \in [0, t_{max}]$, where t_{max} is the maximum number of generations.

Lévy Flight: Increasing Movement and Mimicking Insects

A random walk is the outstanding characteristic of Lévy flight in which the step length follows a Lévy distribution [70]. Equation (8) provides the step length s in Mantegna's algorithm.

$$\text{Levy} \sim s = \frac{u}{|v|^{1/\tau}} \tag{8}$$

where Lévy is a Lévy distribution with an index τ; s denotes a power–law distribution; and u and v are drawn from normal distributions, as follows. New solutions are obtained around the optimal solution using a Lévy walk, which expedites the local search.

$$u \sim N(0,\sigma_u^2), v \sim N(0,\sigma_v^2) \tag{9}$$

$$\text{where } \sigma_u = \left\{ \frac{\Gamma(1+\tau)\sin(\pi\tau/2)}{\Gamma[(1+\tau)/2]\tau 2^{(\tau-1)/2}} \right\}^{1/\tau}, \sigma_v = 1 \tag{10}$$

Here, $\Gamma(\text{t})$ is the Gamma function.

$$\Gamma(\text{t}) = \int_0^\infty z^{t-1}e^{-z}dz \tag{11}$$

Notably, the aforementioned metaheuristic components supplement the basic FA to improve the effectiveness and efficiency of optimization process. The movement of the i^{th} firefly that is attracted to a brighter j^{th} firefly is thus modified as follows:

$$x_i^{t+1} = x_i^t + \beta(x_j^t - x_i^t) + \alpha^t \text{sign}[\text{rand} - 0.5] \otimes \text{Levy} \tag{12}$$

Table 1 presents the default settings of the parameters used in the enhanced FA.

Table 1. Default settings of parameters of enhanced FA.

Group	Parameter	Setting	Purpose
Swarm and metaheuristic settings	Number of fireflies	User defined; default value: 80	Population number
	Max generation	User defined; default value: *tmax* = 40	Constrain implementation of algorithm
	Chaotic logistic map	Random generation; biotic potential η = 4	Generate initial population with high diversity
Brightness	Objective function	Accuracy	Calculate firefly brightness
Attractiveness	βmin	Default value β_0 = 0.1	Minimum value of attractive parameter β
	Chaotic Gauss/mouse map	Random generation	Automatically tune β parameter
	γ	Default value γ = 1	Absorption coefficient
Random movement	α	Default value α_0 = 0.2	Randomness of firefly movement
	Adaptive inertia weight	Default value θ = 0.9	Control the local and global search capabilities of swarm algorithm
	Lévy flight	Default value τ = 1.5	Accelerate the local search by generating new optimal neighborhoods around the obtained best solution

3.2.3. Optimized LSSVM Model with Decomposition Scheme

The hybrid model in this work combines the LSSVM with the OAO decomposition scheme to solve multi-level classification problems. In highly nonlinear spaces, the RBF kernel is used in the LSSVM. To improve accuracy in the solution of multi-class problems, the enhanced FA is used to finetune the regularization parameter (C) and the sigma parameter (σ) in the LSSVM model. Particularly, the FA was improved using three supplementary elements to optimize hyperparameters C and σ. Equation (13) is the fitness function of the model in which the objective function represents the classification accuracy.

$$\text{f(m)} = \text{objective\_function}_{\text{validation-data}} \tag{13}$$

3.3. Performance Measures

3.3.1. Cross-Fold Validation

The k-fold cross-validation technique is extensively applied to confirm the accuracy of algorithms, as it reduces biases that are associated with randomly sampling training and test sets. Kohavi (1995) verified that ten-fold cross-validation was optimal [71]; it involves dividing a complete dataset into ten subsets (nine learning subsets and one test subset).

3.3.2. Confusion Matrix

In the field of machine learning and the problem of statistical classification, the confusion matrix is commonly applied to evaluate the efficacy of an algorithm. Table 2 presents an example of a confusion matrix. From the table, the true positive (*tp*) value and true negative (*tn*) value represent accurate classifications. The false positive (*fp*) value or false negative (*fn*) value refers to erroneous classifications.

Table 2. Confusion matrix.

		Actual Class	
		Positive	**Negative**
Predicted class	**Positive**	True positive (*tp*)	False negative (*fn*)
	Negative	False positive (*fp*)	True negative (*tn*)

The commonly used metrics of the effectiveness of classification are generated from four elements of the confusion matrix (accuracy, precision, sensitivity, specificity and area under the receiver operating characteristic curve (AUC)).

The predictive accuracy of a classification algorithm is calculated as follows.

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{14}$$

Two extended versions of accuracy are precision and sensitivity. Precision measures the reproducibility of a measurement, whereas sensitivity—also called recall—measures the completeness. Precision in Equation (15) is defined as the number of true positives as a proportion of the total number of true positives and false positives that are provided by the classifier.

$$Precision = \frac{tp}{tp + fp} \tag{15}$$

Sensitivity in Equation (16) is the number of correctly classified positive examples divided by the number of positive examples in the data. In identifying positive labels, sensitivity is useful for estimating the effectiveness of a classifier.

$$Sensitivity = \frac{tp}{tp + fn} \tag{16}$$

Another performance metric is specificity. The specificity of a test is the ability of the test to determine correctly those cases. This metric is estimated by calculating the number of true negatives as a proportion of the total number of true negatives and false positives in examples. Equation (17) is the formula for specificity.

$$Specificity = \frac{tn}{tn + fp} \tag{17}$$

A receiver operating characteristic (ROC) curve is the most commonly used tool for visualizing the performance of a classifier, and AUC is the best way to capture its performance as a single number. The ROC curve captures a single point, the area under the curve (AUC), in the analysis of model performance [72]. The AUC, sometimes referred to as the balanced accuracy [73] is easily obtained using Equation (18).

$$AUC = \frac{1}{2}\left[\left(\frac{tp}{tp + fn}\right) + \left(\frac{tn}{tn + fp}\right)\right] \tag{18}$$

4. Metaheuristic-Optimized Multi-Level Classification System

4.1. Benchmarking of the Enhanced Metaheuristic Optimization Algorithm

This section evaluates the efficiency of the enhanced FA by testing benchmark functions to elucidate the characteristics of optimization algorithms. Ten complex benchmark functions with different characteristics and dimensions [74,75] were used herein to evaluate the performance of the enhanced FA. This investigation used 200 for the number of fireflies and 1000 for the maximum number of iterations.

Table 3 presents numerical benchmark functions and their optimal values that are obtained, using the enhanced FA. The results indicate that the enhanced FA yielded all of the optimal values, which were very close to the analytically obtained values. Therefore, the proposed enhanced FA is promising.

Table 3. Numerical benchmark functions.

No.	Benchmark Functions	Dimension	Minimum Value	Maximum Value	Mean of Optimum	Standard Deviation	Total Time (s)
1	Griewank	10	3.03×10^{-11}	3.75×10^{-10}	1.36×10^{-10}	8.44×10^{-11}	2.10×10^{4}
	$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ $x_i = [-600;\ 600]$	30	7.84×10^{-8}	2.36×10^{-7}	1.51×10^{-7}	4.49×10^{-7}	1.99×10^{4}
	Minimum f(0, . . . ,0) = 0	50	5.40×10^{-7}	1.74×10^{-6}	1.17×10^{-6}	3.01×10^{-7}	2.34×10^{4}
2	Deb 01	10	-1	-1	-1	4.98×10^{-12}	1.54×10^{4}
	$f(x) = -\frac{1}{d}\sum_{i=1}^{d} sin^6$(5*pi*x) x_i = [−1;1]	30	-1	-8.34×10^{-1}	-9.93×10^{-1}	3.12×10^{-2}	1.85×10^{4}
	Minimum f(0, . . . ,0) = −1	50	-1	-5.24×10^{-1}	-9.31×10^{-1}	1.39×10^{-1}	2.26×10^{4}
3	Csendes	10	7.04×10^{-11}	1.06×10^{-5}	9.57×10^{-7}	2.09×10^{-6}	3.55×10^{4}
	$f(x) = \sum_{i=1}^{d} x_i^6\left(2 + \sin\frac{1}{x_i}\right)$ x_i = [−1; 1]	30	4.39×10^{-6}	2.39×10^{-3}	5.07×10^{-4}	5.66×10^{-4}	4.27×10^{4}
	Minimum f(0, . . . ,0) = 0	50	3.78×10^{-4}	6.53×10^{-3}	1.49×10^{-3}	1.22×10^{-3}	4.91×10^{4}
4	De Jong	10	2.80×10^{-12}	8.65×10^{-12}	4.82×10^{-12}	1.59×10^{-12}	1.50×10^{4}
	$f(x) = \sum_{i=1}^{d} x_i^2;\ x_i$ = [−5.12; 5.12]	30	7.40×10^{-11}	3.33×10^{-4}	1.11×10^{-5}	6.08×10^{-5}	1.97×10^{4}
	Minimum f(0, . . . ,0) = 0	50	1.39×10^{-4}	4.45×10^{-2}	8.07×10^{-3}	9.66×10^{-3}	2.37×10^{4}
5	Alpine 1	10	6.69×10^{-7}	5.49×10^{-4}	2.03×10^{-5}	9.99×10^{-5}	1.50×10^{4}
	$f(x) = \sum_{i=1}^{d} \lvert x_i \sin(x_i\ ) + 0.1\ x_i\ \rvert$ x_i = [−10; 10]	30	6.80×10^{-6}	7.43×10^{-3}	5.21×10^{-4}	1.65×10^{-3}	2.07×10^{4}
	Minimum f(0, . . . ,0) = 0	50	2.43×10^{-5}	4.95×10^{-3}	9.43×10^{-4}	1.42×10^{-3}	2.34×10^{4}

Table 3. *Cont.*

No.	Benchmark Functions	Dimension	Minimum Value	Maximum Value	Mean of Optimum	Standard Deviation	Total Time (s)
6	Sum Squares	10	4.77×10^{-11}	1.74×10^{-10}	1.06×10^{-10}	3.27×10^{-11}	1.44×10^{4}
	$f(x) = \sum_{i=1}^{d} ix_i^2$ x_i = [−10; 10]	30	1.52×10^{-8}	4.59×10^{-8}	2.70×10^{-8}	7.78×10^{-9}	3.20×10^{4}
	Minimum f(0, . . . ,0) =0	50	1.51×10^{-5}	1.60×10^{-2}	1.13×10^{-3}	2.89×10^{-3}	2.46×10^{4}
7	Rotated hyper-ellipsoid	10	1.96×10^{-9}	6.99×10^{-9}	4.73×10^{-9}	1.29×10^{-9}	1.48×10^{4}
	$f(x) = \sum_{i=1}^{d} \sum_{j=1}^{i} x_j^2$ x_i = [−65.536; 65.536]	30	4.43×10^{-7}	1.56×10^{-6}	1.06×10^{-6}	3.30×10^{-7}	2.40×10^{4}
	Minimum f(0, . . . ,0) = 0	50	3.80×10^{-5}	3.37×10^{-3}	9.75×10^{-4}	1.10×10^{-3}	2.23×10^{4}
8	Xin She Yang 2	10	5.66×10^{-4}	5.66×10^{-4}	5.66×10^{-4}	4.63×10^{-15}	1.59×10^{4}
	$f(x) =$ $\sum_{i=1}^{d} \lvert x_i \rvert$*exp*$[-\sum_{i=1}^{d} \sin(x_i^2)]$ x_i = [−2π; 2π]	30	3.51×10^{-12}	1.06×10^{-11}	5.24×10^{-12}	2.09×10^{-12}	2.32×10^{4}
	Minimum f(0, . . . ,0) = 0	50	4.36×10^{-20}	5.04×10^{-18}	1.18×10^{-18}	1.40×10^{-18}	2.21×10^{4}
9	Schwefel	10	6.36×10^{-58}	1.50×10^{-55}	2.27×10^{-56}	3.28×10^{-56}	3.59×10^{4}
	$f(x) = \sum_{i=1}^{d} x_i^{10};\ x_i$ = [−10; 10]	30	3.42×10^{-49}	4.68×10^{-28}	1.64×10^{-29}	8.55×10^{-29}	4.20×10^{4}
	Minimum f(0, . . . ,0) =0	50	7.08×10^{-18}	3.40×10^{-13}	2.75×10^{-14}	6.75×10^{-14}	4.78×10^{4}
10	Chung-Reynolds	10	3.95×10^{-19}	6.84×10^{-18}	2.74×10^{-18}	1.63×10^{-18}	1.47×10^{4}
	$f(x) = (\sum_{i=1}^{d} x_i^2\ )^2; x_i$ = [−100; 100]	30	1.25×10^{-15}	5.22×10^{-15}	2.22×10^{-15}	9.62×10^{-16}	1.79×10^{4}
	Minimum f(0, . . . ,0) = 0	50	1.99×10^{-14}	1.32×10^{-13}	5.82×10^{-14}	2.74×10^{-14}	2.48×10^{4}

4.2. System Development

The multi-level classification system comprises two computing modules, OAO-LSSVM and Optimized-OAO-LSSVM. Combining the OAO scheme with the LSSVM yielded the baseline model for solving multi-class classification problems. The LSSVM model was then further optimized using a swarm intelligence algorithm (enhanced FA). The GUI was created to help users to be acquainted with the environment of machine learning.

4.2.1. Framework

Figure 1 shows the framework of the proposed multi-level classification system. The two modules of the system are the Optimized-OAO-LSSVM and baseline OAO-LSSVM module. In the system, the users can choose either the Optimized-OAO-LSSVM or baseline OAO-LSSVM module to run the data. Both modules help the user to evaluate model performance or to predict outputs. The system also enables the user to save the model after executing the training process, allowing it to be reused for other purposes.

With the baseline OAO-LSSVM module, the input data are separated into learning data and test data. After setting original input hyperparameters, the learning data help to create the model, and the test data are used to evaluate model or predict output values depending on the demand of users. The main difference between the Optimized-OAO-LSSVM and baseline module is that the input hyperparameters of the Optimized-OAO-LSSVM model are finetuned by the enhanced FA, which improves the performance of the machine learning model.

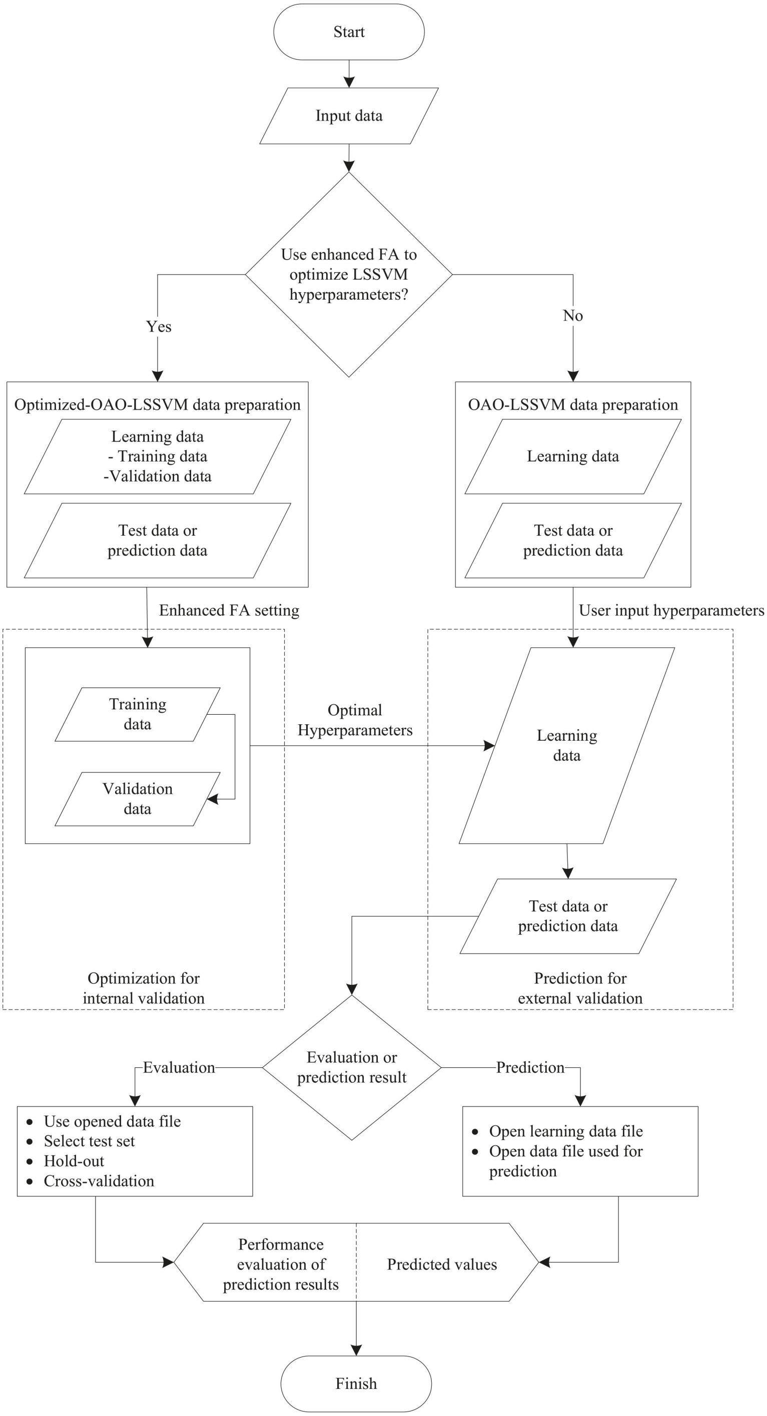
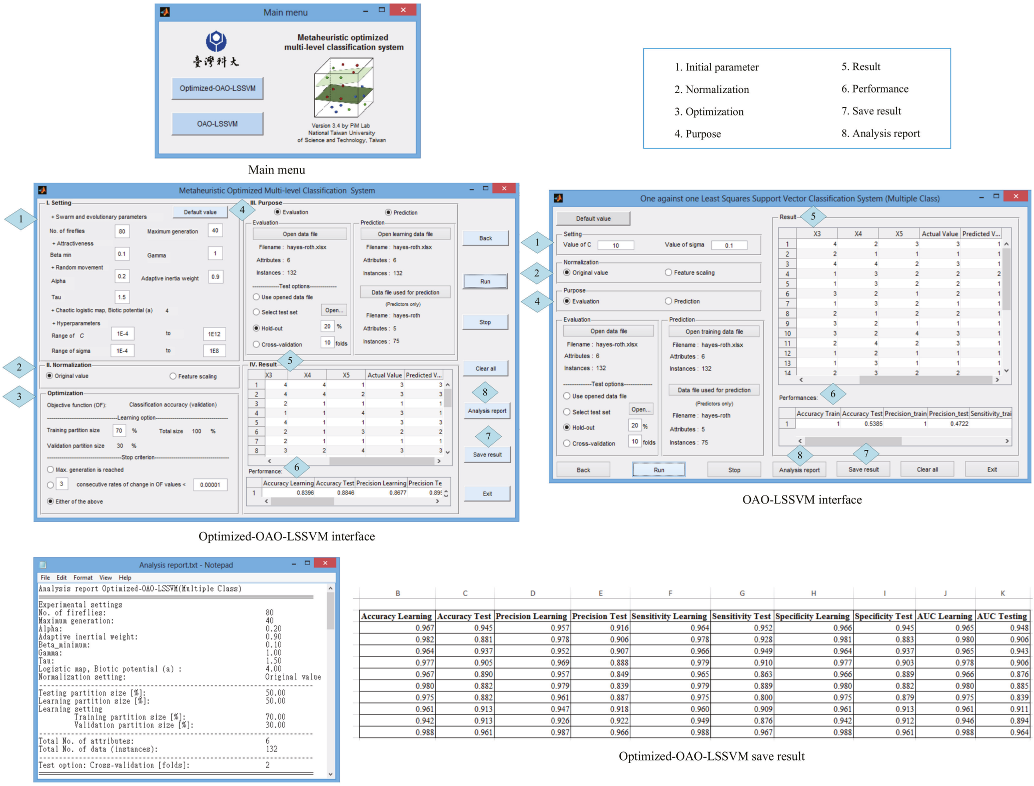

Start
Input data
Use enhanced FA to optimize LSSVM hyperparameters?
Yes
No
Optimized-OAO-LSSVM data preparation
Learning data
- Training data
-Validation data
Test data or prediction data
OAO-LSSVM data preparation
Learning data
Test data or prediction data
Enhanced FA setting
User input hyperparameters
Training data
Validation data
Optimal Hyperparameters
Learning data
Test data or prediction data
Optimization for internal validation
Prediction for external validation
Evaluation or prediction result
Evaluation
Prediction
• Use opened data file
• Select test set
• Hold-out
• Cross-validation
• Open learning data file
• Open data file used for prediction
Performance evaluation of prediction results
Predicted values
Finish

Figure 1. Metaheuristic-optimized multi-level classification system flowchart.

4.2.2. Implementation

The proposed system has two functions, including evaluation and prediction. The evaluation function supports four operations, and users can choose one of these four operations.

Figure 2 shows the screenshots of the system. In the main menu, a user can adopt the enhanced FA to tune the LSSVM hyperparameters. Then, the parameters are set by the user, or the default values are used. Next, the user must select or not select normalization, the part between the training data and the validation data, as well as the stopping criteria.


Main menu
Metaheuristic optimized multi-level classification system
臺灣科大
Optimized-OAO-LSSVM
OAO-LSSVM
Version 3.4 by PiM Lab
National Taiwan University of Science and Technology, Taiwan
Metaheuristic Optimized Multi-level Classification System
I. Setting
II. Normalization
III. Purpose
IV. Result
Optimization
Default value
Open data file
Open learning data file
Data file used for prediction
Back
Run
Stop
Clear all
Analysis report
Save result
Exit
One against one Least Squares Support Vector Classification System (Multiple Class)
1
2
3
4
5
6
7
8

1. Initial parameter
2. Normalization
3. Optimization
4. Purpose
5. Result
6. Performance
7. Save result
8. Analysis report

Main menu

Optimized-OAO-LSSVM interface

OAO-LSSVM interface

```
Analysis report Optimized-OAO-LSSVM(Multiple Class)
==================================================================
Experimental settings
No. of fireflies:                                       80
Maximum generation:                                     40
Alpha:                                                  0.20
Adaptive inertial weight:                               0.90
Beta_minimum:                                           0.10
Gamma:                                                  1.00
Tau:                                                    1.50
Logistic map, Biotic potential (a) :                    4.00
Normalization setting:                                  Original value
------------------------------------------------------------------
Testing partition size [%]:                             50.00
Learning partition size [%]:                            50.00
Learning setting
        Training partition size [%]:                    70.00
        Validation partition size [%]:                  30.00
------------------------------------------------------------------
Total No. of attributes:                                6
Total No. of data (instances):                          132
------------------------------------------------------------------
Test option: Cross-validation [folds]:                  2
==================================================================
```

Optimized-OAO-LSSVM analysis report

B	C	D	E	F	G	H	I	J	K
Accuracy Learning	**Accuracy Test**	**Precision Learning**	**Precision Test**	**Sensitivity Learning**	**Sensitivity Test**	**Specificity Learning**	**Specificity Test**	**AUC Learning**	**AUC Testing**
0.967	0.945	0.957	0.916	0.964	0.952	0.966	0.945	0.965	0.948
0.982	0.881	0.978	0.906	0.978	0.928	0.981	0.883	0.980	0.906
0.964	0.937	0.952	0.907	0.966	0.949	0.964	0.937	0.965	0.943
0.977	0.905	0.969	0.888	0.979	0.910	0.977	0.903	0.978	0.906
0.967	0.890	0.957	0.849	0.965	0.863	0.966	0.889	0.966	0.876
0.980	0.882	0.979	0.839	0.979	0.889	0.980	0.882	0.980	0.885
0.975	0.882	0.961	0.887	0.975	0.800	0.975	0.879	0.975	0.839
0.961	0.913	0.947	0.918	0.960	0.909	0.961	0.913	0.961	0.911
0.942	0.913	0.926	0.922	0.949	0.876	0.942	0.912	0.946	0.894
0.988	0.961	0.987	0.966	0.988	0.967	0.988	0.961	0.988	0.964

Optimized-OAO-LSSVM save result

Figure 2. Screenshots of system.

The results and predicted values obtained by using the Optimized-OAO-LSSVM model are displayed in the interface. Moreover, users can view and save the results as an Excel file, which includes inputs and outputs. The Optimized-OAO-LSSVM system showed the efficiency of operating the proposed model.

5. Engineering Applications

This section elucidates the Optimized-OAO-LSSVM system to handle classification issues. Many case studies in engineering management were used herein to evaluate the application of multi-classification system. Section 5.1 presents the results obtained by using the proposed model to solve binary-class geotechnical problems. Section 5.2 demonstrates the use of the system to solve multi-class civil engineering and construction management problems.

5.1. Binary-Class Problems

Two binary-class datasets associated with seismic hazards in coal mines and the early warning of liquefaction disasters are taken from the literature [76,77]. Table 4 presents the variables and their descriptive statistics of the datasets.

Table 4. Data collection and parameter setting.

Parameter	Unit	Max. Value	Min. Value	Mean	Standard Deviation
Dataset 1—Seismic bumps, 2584 samples, Poland [76]					
Genergy	N/A	2,595,650.00	100.00	90,242.52	229,200.51
Gpuls	N/A	4518.00	2.00	538.58	562.65
Gdenergy	N/A	1245.00	−96.00	12.38	80.32
Gdpuls	N/A	838.00	−96.00	4.51	63.17
Energy	Joule	402,000.00	0.00	4975.27	20,450.83
Maxenergy	Joule	400,000.00	0.00	4278.85	19,357.45
Seismic bumps (1 = hazardous state, 2 = not)	N/A	2	1		
Dataset 2—Soil Liquefaction, 226 samples, U.S.A., China and Taiwan [77]					
Cone tip resistance (q_c)	MPa	25.00	0.90	5.82	4.09
Sleeve friction ratio (R_f)	%	5.20	0.10	1.22	1.05
Effective stress (σ'_v)	kPa	215.20	22.50	74.65	34.40
Total stress (σ_v)	kPa	274.00	26.60	106.89	55.36
Horizontal ground surface acceleration (a_{max})	gal	0.80	0.08	0.29	0.14
Earthquake movement magnitude (M_w)	N/A	7.60	6.00	6.95	0.44
Soil liquefaction (1 = exists, 2 = not)	N/A	2	1		

Note: The users have to convert the output of data into class 1 and 2.

In monitoring seismic hazards in coal mines, an early warning model can be applied to forecast the occurrence of hazard events and withdraw workers from threatened areas, reducing the risk of mechanical seismic impact to save the lives of mine workers. The dataset has 170 samples, representing a hazardous state (Class 1) and 2414 samples, representing a non-hazardous state (Class 2).

Soil liquefaction is a major effect of an earthquake and may seriously damage buildings and infrastructure and cause loss of life. The deformation of soil by a high pore-water pressure causes the liquefaction. A soil deposit under a dynamic load generates pore water, which reduces its strength and causes liquefaction. The proposed model is used to predict the liquefaction or non-liquefaction of soil. This database embraces 226 examples comprising 133 instances of liquefaction (Class 1) and 93 instances of non-liquefaction (Class 2).

Chou et al., (2016) combined the smart firefly algorithm with the LSSVM (SFA-LSSVM) to solve seismic bump and soil liquefaction problems [78]. They compared the performance of the SFA-LSSVM model with the experimental performance of other models and concluded that the SFA-LSSM is the best model for solving such problems.

Therefore, to demonstrate the effectiveness and efficiency of the proposed model in solving binary-class problems, the results obtained using the proposed model were compared with those obtained using the SFA-LSSVM model. Table 5 presents the results of using the Optimized-OAO-LSSVM and SFA-LSSVM models for predicting seismic bumps and soil liquefaction in original-value and feature-scaling cases.

Table 5. Comparison of performances of SFA-LSSVM and Optimized-OAO-LSSVM models used to solve binary problems.

Technique	Cross-Fold Validation	Accuracy (%)	Training and Test Time (s)
Dataset 1—Seismic bumps (2584 samples)			
SFA-LSSVM (original value)	10	93.46	355,913.59
SFA-LSSVM (feature scaling)	10	93.96	174,328.48
Optimized-OAO-LSSVM (original value)	10	93.42	1136.60
Optimized-OAO-LSSVM (feature scaling)	10	93.30	717.37
Dataset 2—Soil liquefaction (226 samples)			
SFA-LSSVM (original value)	10	94.31	19,884.82
SFA-LSSVM (feature scaling)	10	95.18	998.45
Optimized-OAO-LSSVM (original value)	10	93.38	57.22
Optimized-OAO-LSSVM (feature scaling)	10	92.93	56.14

The computational time of the Optimized-OAO-LSSVM model was substantially shorter than that of the SFA-LSSVM model, although its predictive accuracy was not significantly higher. With seismic bumps dataset, the Optimized-OAO-LSSVM model had an accuracy of 93.42% in 1136.60 s whereas the SFA-LSSVM model had an accuracy of 93.46% in 355,913.59 s.

Similarly, the Optimized-OAO-LSSVM model had a shorter computing time than the SFA-LSSVM model with soil liquefaction data (57.22 s and 19,884.82 s with original value case, respectively). Therefore, the Optimized-OAO-LSSVM is an effective and efficient model for solving binary-class classification problems.

5.2. Multi-Level Problems

The proposed system was applied to three multi-level cases. The results obtained were compared with those obtained using the baseline model (OAO-LSSVM), with prior experimental results and with those obtained using single multi-class models (SMO, Multiclass Classifier, Naïve Bayes, Logistic, and LibSVM).

5.2.1. Case 1—Diagnosis of Faults in Steel Plates

Fault diagnosis is important in industrial production. For instance, producing defective products can impose a high cost on a manufacturer of steel products. Therefore, in this investigation a dataset of faults in steel plates, which are important raw materials in hundreds of industrial products, is used as a practical case. The original dataset was obtained from Semeion, Research of Sciences of Communication, Via Sersale 117, 00128, Rome, Italy. In this dataset, faults in steel plates are classified into 7 types, including Pastry, Zscratch, Kscratch, Stains, Dirtiness, Bumps and Other. The database contains 1941 data points with 27 independent variables.

To prevent confusion in multi-class classification, Tian et al. [65] eliminated faults of class 7 because that class did not refer to a particular kind of fault. Furthermore, to improve predictive accuracy, they used the recursive feature elimination (RFE) algorithm to reduce the number of dimensions of the multi-classification. Therefore, Tian et al. used a modified steel plates fault dataset (1268 samples) with 20 independent attributes and six types of fault [65]. To obtain a fair comparison, therefore, the proposed model was applied to the modified data. Table 6 presents the inputs and profile of categorical labels for data concerning faults in steel plates.

Table 6. Statistical input and profile of categorical labels for the steel plate faults diagnosis data.

Parameter	Max. Value	Min. Value	Mean	Standard Deviation
Input				
Edges Y Index	1	0.048	0.813	0.234
Outside Global Index	1	0	0.576	0.482
Orientation Index	1	−0.991	0.083	0.501
Edges X Index	1	0.014	0.611	0.243
Type of Steel_A300	1	0	0.400	0.490
Luminosity Index	1	−0.999	−0.131	0.149
Square Index	1	0.008	0.571	0.271
Type of Steel_A400	1	0	0.600	0.490
Length of Conveyer	1794	1227	1459.160	144.578
Minimum of Luminosity	203	0	84.549	32.134
X Maximum	1713	4	617.964	497.627
X Minimum	1705	0	571.136	520.691
Sigmoid of Areas	1	0.119	0.585	0.339
Edges Index	1	0	0.332	0.300
Empty Index	1	0	0.414	0.137
Maximum of Luminosity	253	37	130.194	18.691
Log of Areas	51,837	0.301	22,757.224	9704.564
Log Y Index	42,587	0	11,636.590	7273.127
Log X Index	30,741	0.301	9477.470	7727.986
Steel Plate Thickness	300	40	78.738	55.086
Output—Type of fault				
Pastry (Class 1)	N/A			
ZScratch (Class 2)				
KScratch (Class 3)				
Stains (Class 4)				
Dirtiness (Class 5)				
Bumps (Class 6)				

Accuracy, precision, sensitivity, specificity and AUC are indices used to evaluate the effectiveness of the proposed model. High values indicate favorable performance and vice versa. Accuracy is the most commonly used index. Table 7 presents the predictive performances of SMO, the Multiclass Classifier, the Naïve Bayes, Logistic, LibSVM and several empirical models [65], and the OAO-LSSVM and Optimized-OAO-LSSVM models when applied to the steel fault dataset.

Tian et al. used three optimizing algorithms—grid search (GS), GA and PSO—combined with SVM to improve the accuracy of classification in the steel fault dataset [65]. They showed that the SVM model, optimized by PSO, was the best for predicting the test data, with an accuracy of 79.6%. With the same data, the Optimized-OAO-LSSVM had an accuracy of 91.085%. The Optimized-OAO-LSSVM model was more accurate than SMO (86.357%), the Multiclass Classifier (85.726%), the Naïve Bayes (82.334%), the Logistic model (86.124%), the LibSVM (31.704%) and the OAO-LSSVM model (53.553%). The statistical accuracy of the Optimized-OAO-LSSVM model, applied to the test data, was better than those of other algorithms at a significance level of 1%.

Table 7. Results of performance measures and rates of improved accuracy achieved by Optimized-OAO-LSSVM.

Dataset	Empirical Models Reported in Primary Works and Single Multi-Class Models	Performance Measure					Improved Accuracy by Optimized-OAO-LSSVM System (%)
		Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	AUC	
Dataset 1—Diagnosis of faults in steel plates	SMO	86.357	86.400	86.300	95.300	0.908	5.191
	Multiclass Classifier	85.726	85.700	85.600	96.000	0.908	5.884
	Naïve Bayes	82.334	82.300	84.440	95.960	0.902	9.608
	Logistic	86.124	86.100	86.000	97.400	0.917	5.447
	LibSVM	31.704	31.700	10.100	89.900	0.500	65.193
	GS-SVM [65]	77.800	-	-	-	-	14.586
	GA-SVM [65]	78.000	-	-	-	-	14.366
	PSO-SVM [65]	79.600	-	-	-	-	12.610
	OAO-LSSVM	53.553	28.764	-	59.148	-	41.206
	Optimized-OAO-LSSVM	91.085	89.995	90.437	91.020	0.907	-
Dataset 2—Quality of water in reservoir	SMO	75.238	75.200	77.500	85.900	0.817	19.661
	Multiclass Classifier	85.397	85.400	86.500	94.900	0.907	8.813
	Naïve Bayes	76.000	76.000	78.700	99.500	0.891	18.847
	Logistic	89.580	89.600	89.600	95.000	0.923	4.346
	LibSVM	80.950	81.000	81.000	87.600	0.843	13.561
	OAO-LSSVM	92.196	90.794	90.633	92.078	0.914	1.553
	Optimized-OAO-LSSVM	93.650	92.531	93.840	93.746	0.938	-
Dataset 3—Urban land cover	SMO	85.778	85.800	86.000	89.000	0.875	1.714
	Multiclass Classifier	64.900	64.900	64.800	99.400	0.821	25.636
	Naïve Bayes	81.000	81.000	81.600	91.800	0.867	7.189
	Logistic	65.926	65.900	65.900	95.300	0.806	24.461
	LibSVM	18.370	18.400	19.000	81.400	0.502	78.951
	k-NN classifier [79]	80.140	-	-	-	-	8.174
	ELM classifier [79]	84.700	-	-	-	-	2.949
	SVM classifier [79]	84.890	-	-	-	-	2.732
	OAO-LSSVM	18.378	11.637	-	-	-	78.942
	Optimized-OAO-LSSVM	87.274	87.048	89.918	87.297	0.886	-

5.2.2. Case 2—Quality of Water in Reservoir

The case study from the field of hydroelectric engineering involves a dataset on the quality of water in a reservoir. The quality of water is critical because water is a primary natural resource that supports the survival and health of humans through drinking, irrigation, hydroelectricity, aquaculture and recreation. Accurately predicting water quality is essential in the management of water resources.

Table 8 shows the details of the water quality dataset. Carlson's Trophic State Index (CTSI) has long been used in Taiwan to assess eutrophication in reservoirs [80]. Generally, the factors that are considered to evaluate reservoir water quality are quite complex. The key assessment factors include Secchi disk depth (SD), chlorophyll a (Chla), total phosphorus (TP), dissolved oxygen (DO), ammonia (NH3), biochemical oxygen demand (BOD), temperature (TEMP) and others. In this investigation, SD, Chla and TP were used to classify the quality of water in a reservoir. The OECD's single indicator water quality differentiations (Table 9) [81] was used to generate the following five levels for each evaluation factor, as follows; excellent (Class 1), good (Class 2), average (Class 3), fair (Class 4) and poor (Class 5). The database includes 1576 data points with three independent inputs (SD, Chla and TP) and the output is one of five ratings of quality of water in a reservoir.

Table 8. Statistical attributes of reservoir water quality dataset.

Parameter	Max. Value	Min. Value	Mean	Standard Deviation
Input				
Secchi disk depth (SD)	8.375	0.1	1.8605	1.1026
Chlorophyll a (Chla)	151.4	0.1	7.9216	12.2305
Total phosphorus (TP)	2.0495	0.0022	0.0677	0.214
Output-Reservoir water quality Excellent—Class 1 Good—Class 2 Average—Class 3 Fair—Class 4 Poor—Class 5	N/A			

Table 9. Single indicator water quality differentiations.

Factor/Index	Excellent 1	Good 2	Average 3	Fair 4	Poor 5
Secchi disk depth (SD)	>4.5	4.5–3.7	3.7–2.3	2.3–1.7	<1.7
Chlorophyll a (Chla)	<2	2.0–3.0	3.0–7.0	7.0–10.0	>10
Total phosphorus (TP)	<8	8–12	12–28	28–40	>40
Carlon's Trophic State Index (CTSI)	<20	20–40	40–50	50–70	>70

Table 7 compares the performances of the SMO, Multiclass Classifier, Naïve Bayes, Logistic, LibSVM, OAO-LSSVM and Optimized-OAO-LSSVM models when used to predict the quality of water in a reservoir, using test data. The numerical results revealed that the Optimized-OAO-LSSVM is the best model for predicting this dataset in terms of accuracy, precision, sensitivity, specificity and AUC value (93.650% 92.531%, 93.840%, 93.746% and 0.938 respectively). Moreover, the hypothesis tests concerning accuracy established that the Optimized-OAO-LSSVM model was more efficient than the other models at a significance level of 1%.

5.2.3. Case 3—Urban Land Cover

Another dataset, concerning urban land cover (675 data points), was obtained from the UCI Machine Learning Repository [82]. Information about land use is important in every city because it is used for many purposes [83], including tax assessment, setting land use policy, city planning, zoning regulation, analysis of environmental processes, and management of natural resources. The assessment of land cover is very important for scientists and authorities that are concerned with mapping the patterns of land cover on global, regional as well as local scales, to understand geographical changes [79]. Therefore, accurate and readily produced land cover classification maps are of great importance in studies of global change.

The land cover dataset includes a total of 147 features, which include the spectral, magnitude, formal and textural properties of an image of land. The spectral, magnitude, formal and textural properties of the image consist of 21 features. Afterwards, these features were repeated on each coarse scales (20, 40, 60, 80, 100, 120, and 140), yielding 147 features [79]. Table 10 shows the features used in the dataset. The data specify nine forms of land cover—trees (Class 1), concrete (Class 2), shadows (Class 3), asphalt (Class 4), buildings (Class 5), grass (Class 6), pools (Class 7), cars (Class 8) and soil (Class 9)—which are treated as the predictive classes, and listed in Table 11.

Table 10. Attribute information in the urban land cover dataset.

Names of Attributes in the Dataset	Source of Information of the Segments
BrdIndx: border index	Shape
Area: area in m^2	Size
Round: roundness	Shape
Bright: brightness	Spectral
Compact: compactness	Shape
ShpIndx: shape index	Shape
Mean_G: green	Spectral
Mean_R: red	Spectral
Mean_NIR: near Infrared	Spectral
SD_G: standard deviation of green	Texture
SD_R: standard deviation of red	Texture
SD_NIR: standard deviation of near infrared	Texture
LW: length/width	Shape
GLCM1: gray-level co-occurrence matrix	Texture
Rect: rectangularity	Shape
GLCM2: another gray-level co-occurrence matrix attribute	Texture
Dens: density	Shape
Assym: asymmetry	Shape
NDVI: normalized difference vegetation index	Spectral
BordLngth: border length	Shape
GLCM3: another gray-level co-occurrence matrix attribute	Texture

Note: These attributes are repeated for each coarse scale (i.e., variable_20, variable_40 . . . , variable_140).

Table 11. Number of data points concerning nine forms of land cover in urban land cover dataset.

Names of the Land Cover in the Dataset	No. of Data Points
Trees (Class 1)	106
Concrete (Class 2)	122
Shadow (Class 3)	61
Asphalt (Class 4)	59
Buildings (Class 5)	112
Grass (Class 6)	116
Pools (Class 7)	29
Cars (Class 8)	36
Soil (Class 9)	34
Total	675

Durduran [79] used three classification algorithms, k-NN, SVM and extreme learning machine (ELM), each combined with the OAR scheme, to predict urban land cover. To verify the effectiveness of the proposed Optimized-OAO-LSSVM model in classifying urban land cover, the performance of the proposed model is compared with their experimental results.

Table 7 compares the predictive accuracies of the SMO, Multiclass Classifier, Naïve Bayes, Logistic, LibSVM, OAO-LSSVM, and the proposed models with that, experimentally determined, of k-NN, SVM, and ELM. As shown in Table 10, the Optimized-OAO-LSSVM had an accuracy of 87.274%, a precision of 87.048%, a sensitivity of 89.918%, a specificity of 87.297% and an AUC of 0.886. Clearly, the Optimized-OAO-LSSVM model outperformed the other models in all these respects. Notably, the Optimized-OAO-LSSVM model is more efficient than the other models at a significance level of 1%.

5.3. Analytical Results and Discussion

The performance of the proposed classification system was evaluated in terms of accuracy, precision, sensitivity, specificity and AUC. High values of these indices revealed favorable performance and vice versa. However, accuracy is the most commonly used for comparison. Table 7 summarizes the values of the performance metrics in case studies 1–3.

The applicability and efficiency of the proposed system were confirmed by comparing its performance with other single multi-class and previous models.

Data preprocessing, such as data cleansing and transformation, is essential to improving the results of data analysis [84]. The user can decide whether or not to normalize data to the (0, 1) range. Normalizing a dataset can minimize the effect of scaling. Table 12 presents the results of applying the proposed system in the three case studies with the original data and the data after feature scaling. In Table 12, better predictive accuracies were obtained with the original steel plates fault and land cover datasets (91.085% and 87.274%, respectively), whereas better results were obtained with the reservoir water quality dataset after feature scaling (93.650%).

Table 12. Analytical results obtained using Optimized-OAO-LSSVM.

Dataset	Performance Measure				
	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	AUC
Dataset 1—Diagnosis of faults in steel plates					
Original value	91.085	89.995	90.437	91.020	0.907
Feature scaling	88.646	86.518	88.458	88.620	0.885
Dataset 2—Quality of water in reservoir					
Original value	93.526	92.335	94.272	93.622	0.939
Feature scaling	93.650	92.531	93.840	93.746	0.938
Dataset 3—Urban land cover					
Original value	87.274	87.048	89.918	87.297	0.886
Feature scaling	86.521	86.003	87.310	86.534	0.874

6. Conclusions and Recommendation

This work proposed a hybrid inference model that integrated an enhanced firefly algorithm (enhanced FA) with a least squares support vector machine (LSSVM) model and decomposition strategy (i.e., one-against-one, OAO) to improve its predictive accuracy in solving multi-level classification problems. The proposed system provides a baseline classification model, called OAO-LSSVM. The effectiveness of the enhanced FA Optimized-OAO-LSSVM model is compared with that of the baseline OAO-LSSVM model.

To verify the applicability and efficiency of the proposed model in solving multi-level classification problems, the predictive performance of the model was compared to other multi-classification methods and prior studies with respect to accuracy, precision, sensitivity, specificity and AUC. Three case studies, involving the multi-class problems of categorizing steel plate faults, assessing the water quality in a reservoir, and managing the condition of urban land cover, were considered. The proposed model exhibited higher predictive accuracy than the baseline model (OAO-LSSVM), experimental studies and other single multi-class algorithms with the highest accuracy in each case. In particular, the proposed model yielded 91.085%, 93.650% and 87.274% accuracy in steel plate faults, water quality in a reservoir, and urban land cover, respectively. Therefore, the model can be used as a decision-making tool in solving practical problems in the fields of civil engineering and construction management.

A main contribution of this work is the extension of a binary-class model to a meta-heuristically optimized multi-level model for efficiently and effectively solving classification problems involving multi-class data. Another major contribution is the design of an intelligent computing system for users with ease that was proved to be an effective project management software. Although the proposed model exhibited excellent predictive accuracy, and a graphical user interface was effectively implemented, it has limitations that should be addressed by future studies. The proposed model does not have high predictive accuracy when applied to small datasets or the unbalanced numbers of data points. Future studies should also improve the model to make it useful for solving multiple inputs and multiple outputs of multiclass classification problems, and develop it in a cloud computing environment to increase its ubiquitous applicability.

Author Contributions: Conceptualization, J.-S.C.; data curation, T.T.P.P. and C.-C.H.; formal analysis, J.-S.C. and T.T.P.P.; funding acquisition, J.-S.C.; investigation, J.-S.C. and T.T.P.P.; methodology, J.-S.C. and T.T.P.P.; project administration, J.-S.C. and C.-C.H.; resources, J.-S.C. and C.-C.H.; software, J.-S.C. and T.T.P.P.; supervision, J.-S.C.; validation, J.-S.C., T.T.P.P. and C.-C.H.; visualization, J.-S.C. and T.T.P.P.; writing—original draft, J.-S.C. and T.T.P.P.; writing—review and editing, J.-S.C. and T.T.P.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan, under grants 108-2221-E-011-003-MY3 and 107-2221-E-011-035-MY3.

Data Availability Statement: The data that support the findings of this study are available from the UCI Machine Learning Repository and corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the Ministry of Science and Technology, Taiwan, for financially supporting this research.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Sesmero, M.P.; Alonso-Weber, J.M.; Gutierrez, G.; Ledezma, A.; Sanchis, A. An ensemble approach of dual base learners for multi-class classification problems. *Inf. Fusion* **2015**, *24*, 122–136. [CrossRef]
2. Khaledian, Y.; Miller, B.A. Selecting appropriate machine learning methods for digital soil mapping. *Appl. Math. Model.* **2020**, *81*, 401–418. [CrossRef]
3. Wang, Q.; Li, Q.; Wu, D.; Yu, Y.; Tin-Loi, F.; Ma, J.; Gao, W. Machine learning aided static structural reliability analysis for functionally graded frame structures. *Appl. Math. Model.* **2020**, *78*, 792–815. [CrossRef]
4. Bhardwaj, A.; Tiwari, A. Breast cancer diagnosis using Genetically Optimized Neural Network model. *Expert Syst. Appl.* **2015**, *42*, 4611–4620. [CrossRef]
5. Iounousse, J.; Er-Raki, S.; El Motassadeq, A.; Chehouani, H. Using an unsupervised approach of Probabilistic Neural Network (PNN) for land use classification from multitemporal satellite images. *Appl. Soft Comput.* **2015**, *30*, 1–13. [CrossRef]
6. Huang, N.; Xu, D.; Liu, X.; Lin, L. Power quality disturbances classification based on S-transform and probabilistic neural network. *Neurocomputing* **2012**, *98*, 12–23. [CrossRef]
7. Del Vecchio, C.; Fenu, G.; Pellegrino, F.A.; Di Foggia, M.; Quatrale, M.; Benincasa, L.; Iannuzzi, S.; Acernese, A.; Correra, P.; Glielmo, L. Support Vector Representation Machine for superalloy investment casting optimization. *Appl. Math. Model.* **2019**, *72*, 324–336. [CrossRef]
8. Kaefer, F.; Heilman, C.M.; Ramenofsky, S.D. A neural network application to consumer classification to improve the timing of direct marketing activities. *Comput. Oper. Res.* **2005**, *32*, 2595–2615. [CrossRef]
9. Karakatič, S.; Podgorelec, V. Improved classification with allocation method and multiple classifiers. *Inf. Fusion* **2016**, *31*, 26–42. [CrossRef]
10. Wang, Z.Z.; Zhang, Z. Seismic damage classification and risk assessment of mountain tunnels with a validation for the 2008 Wenchuan earthquake. *Soil Dyn. Earthq. Eng.* **2013**, *45*, 45–55. [CrossRef]
11. Chou, J.-S.; Hsu, S.-C.; Lin, C.-W.; Chang, Y.-C. Classifying Influential Information to Discover Rule Sets for Project Disputes and Possible Resolutions. *Int. J. Proj. Manag.* **2016**, *34*, 1706–1716. [CrossRef]
12. Hsu, C.-W.; Lin, C.-J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425. [CrossRef] [PubMed]
13. Lorena, A.C.; de Carvalho, A.C.P.L.F.; Gama, J.M.P. A review on the combination of binary classifiers in multiclass problems. *Artif. Intell. Rev.* **2009**, *30*, 19. [CrossRef]
14. Pal, M. Multiclass Approaches for Support Vector Machine Based Land Cover Classification. *arXiv* **2008**, arXiv:0802.2411, 1–16.
15. Rifkin, R.; Klautau, A. In Defense of One-Vs-All Classification. *J. Mach. Learn. Res.* **2004**, *5*, 101–141.
16. Galar, M.; Fernández, A.; Barrenechea, E.; Bustince, H.; Herrera, F. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognit.* **2011**, *44*, 1761–1776. [CrossRef]
17. Perner, P.; Vingerhoeds, R. Special issue data mining and machine learning. *Eng. Appl. Artif. Intell.* **2009**, *22*, 1–2. [CrossRef]
18. Vapnik, V.N. *Statistical Learning Theory*; John Wiley and Sons: New York, NY, USA, 1998.
19. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.* **2009**, *11*, 10–18. [CrossRef]
20. Galar, M.; Fernández, A.; Barrenechea, E.; Herrera, F. DRCW-OVO: Distance-based relative competence weighting combination for One-vs-One strategy in multi-class problems. *Pattern Recognit.* **2015**, *48*, 28–42. [CrossRef]
21. Kang, S.; Cho, S.; Kang, P. Constructing a multi-class classifier using one-against-one approach with different binary classifiers. *Neurocomputing* **2015**, *149*, 677–682. [CrossRef]

22. Balazs, J.A.; Velásquez, J.D. Opinion Mining and Information Fusion: A survey. *Inf. Fusion* **2016**, *27*, 95–110. [CrossRef]
23. Medhat, W.; Hassan, A.; Korashy, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Eng. J.* **2014**, *5*, 1093–1113. [CrossRef]
24. Yang, X.; Yu, Q.; He, L.; Guo, T. The one-against-all partition based binary tree support vector machine algorithms for multi-class classification. *Neurocomputing* **2013**, *113*, 1–7. [CrossRef]
25. Kim, K.-j.; Ahn, H. A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. *Comput. Oper. Res.* **2012**, *39*, 1800–1811. [CrossRef]
26. Chou, J.-S.; Pham, A.-D. Nature-inspired metaheuristic optimization in least squares support vector regression for obtaining bridge scour information. *Inf. Sci.* **2017**, *399*, 64–80. [CrossRef]
27. Van Gestel, T.; Suykens, J.A.K.; Baesens, B.; Viaene, S.; Vanthienen, J.; Dedene, G.; de Moor, B.; Vandewalle, J. Benchmarking Least Squares Support Vector Machine Classifiers. *Mach. Learn.* **2004**, *54*, 5–32. [CrossRef]
28. Parouha, R.P.; Das, K.N. An efficient hybrid technique for numerical optimization and applications. *Comput. Ind. Eng.* **2015**, *83*, 193–216. [CrossRef]
29. Hanafi, R.; Kozan, E. A hybrid constructive heuristic and simulated annealing for railway crew scheduling. *Comput. Ind. Eng.* **2014**, *70*, 11–19. [CrossRef]
30. Setak, M.; Feizizadeh, F.; Tikani, H.; Ardakani, E.S. A bi-level stochastic optimization model for reliable supply chain in competitive environments: Hybridizing exact method and genetic algorithm. *Appl. Math. Model.* **2019**, *75*, 310–332. [CrossRef]
31. Dabiri, N.; Tarokh, M.J.; Alinaghian, M. New mathematical model for the bi-objective inventory routing problem with a step cost function: A multi-objective particle swarm optimization solution approach. *Appl. Math. Model.* **2017**, *49*, 302–318. [CrossRef]
32. Janardhanan, M.N.; Li, Z.; Bocewicz, G.; Banaszak, Z.; Nielsen, P. Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times. *Appl. Math. Model.* **2019**, *65*, 256–270. [CrossRef]
33. Zhang, J.; Xiao, M.; Gao, L.; Pan, Q. Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. *Appl. Math. Model.* **2018**, *63*, 464–490. [CrossRef]
34. Qingjie, L.; Guiming, C.; Xiaofang, L.; Qing, Y. Genetic algorithm based SVM parameter composition optimization. *Comput. Appl. Softw.* **2012**, *4*, 29.
35. Rastegar, S.; Araújo, R.; Mendes, J. Online identification of Takagi–Sugeno fuzzy models based on self-adaptive hierarchical particle swarm optimization algorithm. *Appl. Math. Model.* **2017**, *45*, 606–620. [CrossRef]
36. Pal, S.K.; Rai, C.S.; Singh, A.P. Comparative study of firefly algorithm and particle swarm optimization for noisy non-linear optimization problems. *Int. J. Intell. Syst. Appl.* **2012**, *4*, 50–57. [CrossRef]
37. Olamaei, J.; Moradi, M.; Kaboodi, T. A new adaptive modified Firefly Algorithm to solve optimal capacitor placement problem. In Proceedings of the 18th Electric Power Distribution Conference, Kermanshah, Iran, 30 April–1 May 2013; pp. 1–6.
38. Chou, J.-S.; Pham, A.-D. Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength. *Constr. Build. Mater.* **2013**, *49*, 554–563. [CrossRef]
39. Coelho, L.d.S.; Mariani, V.C. Improved firefly algorithm approach applied to chiller loading for energy conservation. *Energy Build.* **2013**, *59*, 273–278. [CrossRef]
40. Chou, J.-S.; Lin, C. Predicting Disputes in Public-Private Partnership Projects: Classification and Ensemble Models. *J. Comput. Civ. Eng.* **2013**, *27*, 51–60. [CrossRef]
41. Cheng, M.-Y.; Hoang, N.-D. Risk Score Inference for Bridge Maintenance Project Using Evolutionary Fuzzy Least Squares Support Vector Machine. *J. Comput. Civ. Eng.* **2014**, *28*, 04014003. [CrossRef]
42. Pant, P.; Chatterjee, D. Prediction of clad characteristics using ANN and combined PSO-ANN algorithms in laser metal deposition process. *Surf. Interfaces* **2020**, *21*, 100699. [CrossRef]
43. Noori, N.; Kalin, L.; Isik, S. Water quality prediction using SWAT-ANN coupled approach. *J. Hydrol.* **2020**, *590*, 125220. [CrossRef]
44. Sujith, M.; Jeon, J.-S. Machine Learning–Based Failure Mode Recognition of Circular Reinforced Concrete Bridge Columns: Comparative Study. *J. Struct. Eng.* **2019**, *145*, 04019104.
45. Vapnik, V.N. *The Nature of Statistical Learning Theory*; Springer: Berlin/Heidelberg, Germany, 1995.
46. Liu, Y.; Wang, H.; Zhang, H.; Liber, K. A comprehensive support vector machine-based classification model for soil quality assessment. *Soil Tillage Res.* **2016**, *155*, 19–26. [CrossRef]
47. Gholami, R.; Rasouli, V.; Alimoradi, A. Improved RMR Rock Mass Classification Using Artificial Intelligence Algorithms. *Rock Mech. Rock Eng.* **2013**, *46*, 1199–1209. [CrossRef]
48. Liao, Y.; Xu, J.; Wang, W. A Method of Water Quality Assessment Based on Biomonitoring and Multiclass Support Vector Machine. *Procedia Environ. Sci.* **2011**, *10*, 451–457. [CrossRef]
49. Du, S.; Liu, C.; Xi, L. A Selective Multiclass Support Vector Machine Ensemble Classifier for Engineering Surface Classification Using High Definition Metrology. *J. Manuf. Sci. Eng.* **2015**, *137*, 011003. [CrossRef]
50. Polat, K.; Güneş, S. A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems. *Expert Syst. Appl.* **2009**, *36*, 1587–1592. [CrossRef]
51. Garcia, L.P.F.; Sáez, J.A.; Luengo, J.; Lorena, A.C.; de Carvalho, A.C.P.L.F.; Herrera, F. Using the One-vs-One decomposition to improve the performance of class noise filters via an aggregation strategy in multi-class classification problems. *Knowl. Based Syst.* **2015**, *90*, 153–164. [CrossRef]

52. Zhou, L.; Tam, K.P.; Fujita, H. Predicting the listing status of Chinese listed companies with multi-class classification models. *Inf. Sci.* **2016**, *328*, 222–236. [CrossRef]
53. Suykens, J.A.K.; Gestel, T.V.; Brabanter, J.D.; Moor, B.D.; Vandewalle, J. *Least Squares Support Vector Machines*; World Scientific: Singapore, 2002.
54. Khemchandani, R.; Jayadeva; Chandra, S. Regularized least squares fuzzy support vector regression for financial time series forecasting. *Expert Syst. Appl.* **2009**, *36*, 132–138. [CrossRef]
55. Haifeng, W.; Dejin, H. Comparison of SVM and LS-SVM for Regression. In Proceedings of the 2005 International Conference on Neural Networks and Brain, Beijing, China, 13–15 October 2005; pp. 279–283.
56. Yang, X.-S. *Firefly Algorithm*; Luniver Press: Bristol, UK, 2008.
57. Banati, H.; Bajaj, M. Fire Fly Based Feature Selection Approach. *Int. J. Comput. Sci. Issues* **2011**, *8*, 473–479.
58. Fister, I.; Fister, I., Jr.; Yang, X.-S.; Brest, J. A comprehensive review of firefly algorithms. *Swarm Evol. Comput.* **2013**, *13*, 34–46. [CrossRef]
59. Khadwilard, A.; Chansombat, S.; Thepphakorn, T.; Thapatsuwan, P.; Chainate, W.; Pongcharoen, P. Application of Firefly Algorithm and Its Parameter Setting for Job Shop Scheduling. *J. Ind. Technol.* **2012**, *8*, 49–58.
60. Aungkulanon, P.; Chai-ead, N.; Luangpaiboon, P. Simulated Manufacturing Process Improvement via Particle Swarm Optimisation and Firefly Algorithms. In *Lectures Notes Engineering and Computer Science*; Newswood Limited: Hong Kong, China, 2011.
61. Aci, M.; İnan, C.; Avci, M. A hybrid classification method of k nearest neighbor, Bayesian methods and genetic algorithm. *Expert Syst. Appl.* **2010**, *37*, 5061–5067. [CrossRef]
62. Chou, J.-S.; Cheng, M.-Y.; Wu, Y.-W. Improving classification accuracy of project dispute resolution using hybrid artificial intelligence and support vector machine models. *Expert Syst. Appl.* **2013**, *40*, 2263–2274. [CrossRef]
63. Lee, Y.; Lee, J. Binary tree optimization using genetic algorithm for multiclass support vector machine. *Expert Syst. Appl.* **2015**, *42*, 3843–3851. [CrossRef]
64. Seera, M.; Lim, C.P. A hybrid intelligent system for medical data classification. *Expert Syst. Appl.* **2014**, *41*, 2239–2249. [CrossRef]
65. Tian, Y.; Fu, M.; Wu, F. Steel plates fault diagnosis on the basis of support vector machines. *Neurocomputing* **2015**, *151*, 296–303. [CrossRef]
66. García-Pedrajas, N.; Ortiz-Boyer, D. An empirical study of binary classifier fusion methods for multiclass classification. *Inf. Fusion* **2011**, *12*, 111–130. [CrossRef]
67. Yang, X.-S. *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: Beckington, UK, 2008; p. 128.
68. Hong, W.-C.; Dong, Y.; Chen, L.-Y.; Wei, S.-Y. SVR with hybrid chaotic genetic algorithms for tourism demand forecasting. *Appl. Soft Comput.* **2011**, *11*, 1881–1890. [CrossRef]
69. Gandomi, A.H.; Yang, X.S.; Talatahari, S.; Alavi, A.H. Firefly algorithm with chaos. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 89–98. [CrossRef]
70. Li, X.; Niu, P.; Liu, J. Combustion optimization of a boiler based on the chaos and Lévy flight vortex search algorithm. *Appl. Math. Model.* **2018**, *58*, 3–18. [CrossRef]
71. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th International Joint Conference on Artificial Intelligence—Volume 2, Montreal, QC, Canada, 20 August 1995; pp. 1137–1143.
72. Chou, J.-S.; Tsai, C.-F.; Lu, Y.-H. Project dispute prediction by hybrid machine learning techniques. *J. Civ. Eng. Manag.* **2013**, *19*, 505–517. [CrossRef]
73. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
74. Jamil, M.; Yang, X.-S. A Literature Survey of Benchmark Functions For Global Optimization Problems. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 150–194.
75. Surjanovic, S.; Bingham, D. Virtual Library of Simulation Experiments: Test Functions and Datasets. Available online: http://www.sfu.ca/~{}ssurjano/optimization.html (accessed on 8 May 2016).
76. Sikora, M.; Wróbel, Ł. Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines. *Arch. Min. Sci.* **2010**, *55*, 91–114.
77. Goh, A.T.C.; Goh, S.H. Support vector machines: Their use in geotechnical engineering as illustrated using seismic liquefaction data. *Comput. Geotech.* **2007**, *34*, 410–421. [CrossRef]
78. Chou, J.-S.; Thedja, J.P.P. Metaheuristic optimization within machine learning-based classification system for early warnings related to geotechnical problems. *Autom. Constr.* **2016**, *68*, 65–80. [CrossRef]
79. Durduran, S.S. Automatic classification of high resolution land cover using a new data weighting procedure: The combination of k-means clustering algorithm and central tendency measures (KMC–CTM). *Appl. Soft Comput.* **2015**, *35*, 136–150. [CrossRef]
80. Chou, J.-S.; Ho, C.-C.; Hoang, H.-S. Determining quality of water in reservoir using machine learning. *Ecol. Inform.* **2018**, *44*, 57–75. [CrossRef]
81. Hydrotech Research Institute of National Taiwan University. Reservoir Eutrophiction Prediction and Prevention by Using Remote Sensing Technique. Water Resources Agency: Taipei, Taiwan, 2005. (In Chinese)
82. Lichman, M. *UCI Machine Learning Repository*; University of California, School of Information and Computer Science: Irvine, CA, USA, 2013. Available online: http://archive.ics.uci.edu/ml (accessed on 20 December 2015).

83. Wentz, E.A.; Stefanov, W.L.; Gries, C.; Hope, D. Land use and land cover mapping from diverse data sources for an arid urban environments. *Comput. Environ. Urban Syst.* **2006**, *30*, 320–346. [CrossRef]

84. Crone, S.; Guajardo, J.; Weber, R. The impact of preprocessing on support vector regression and neural networks in time series prediction. In Proceedings of the International Conference on Data Mining DMIN'06, Las Vegas, NV, USA, 26–29 June 2006; pp. 37–42.

symmetry

MDPI

Article

Evolving Hybrid Cascade Neural Network Genetic Algorithm Space–Time Forecasting

Rezzy Eko Caraka [1,2,*], Hasbi Yasin [3], Rung-Ching Chen [4,*], Noor Ell Goldameir [5], Budi Darmawan Supatmanto [6], Toni Toharudin [7], Mohammad Basyuni [8,*], Prana Ugiana Gio [9] and Bens Pardamean [2,10]

[1] Faculty of Economics and Business, Campus UI Depok, Universitas Indonesia, Depok 16426, Indonesia
[2] Bioinformatics and Data Science Research Center, Bina Nusantara University, Jakarta 11480, Indonesia; bpardamean@binus.edu
[3] Department of Statistics, Diponegoro University, Semarang 50275, Indonesia; hasbiyasin@live.undip.ac.id
[4] Department of Information Management, College of Informatics, Chaoyang University of Technology, Taichung 41349, Taiwan
[5] Department of Mathematics, Riau University, Pekanbaru 28293, Indonesia; noorellgoldameir@lecturer.unri.ac.id
[6] Weather Modification Technology Center, Agency for the Assessment and Application of Technology (BPPT), Jakarta 10340, Indonesia; budi.darmawan@bppt.go.id
[7] Department of Statistics, Padjadjaran University, Bandung 16426, Indonesia; toni.toharudin@unpad.ac.id
[8] Department of Forestry, Faculty of Forestry, Universitas Sumatera Utara, Medan 20155, Indonesia
[9] Department of Mathematics, Universitas Sumatera Utara, Medan 20155, Indonesia; prana@usu.ac.id
[10] Computer Science Department, Bina Nusantara University, Jakarta 11480, Indonesia
* Correspondence: rezzy.eko@ui.ac.id (R.E.C.); crching@cyut.edu.tw (R.-C.C.); m.basyuni@usu.ac.id (M.B.)

Abstract: Design: At the heart of time series forecasting, if nonlinear and nonstationary data are analyzed using traditional time series, the results will be biased. At the same time, if just using machine learning without any consideration given to input from traditional time series, not much information can be obtained from the results because the machine learning model is a black box. Purpose: In order to better study time series forecasting, we extend the combination of traditional time series and machine learning and propose a hybrid cascade neural network considering a metaheuristic optimization genetic algorithm in space–time forecasting. Finding: To further show the utility of the cascade neural network genetic algorithm, we use various scenarios for training and testing while also extending simulations by considering the activation functions SoftMax, radbas, logsig, and tribas on space–time forecasting of pollution data. During the simulation, we perform numerical metric evaluations using the root-mean-square error (RMSE), mean absolute error (MAE), and symmetric mean absolute percentage error (sMAPE) to demonstrate that our models provide high accuracy and speed up time-lapse computing.

Keywords: cascade neural network; space–time; forecasting; genetic algorithm; particulate matter

Citation: Caraka, R.E.; Yasin, H.; Chen, R.-C.; Goldameir, N.E.; Supatmanto, B.D.; Toharudin, T.; Basyuni, M.; Gio, P.U.; Pardamean, B. Evolving Hybrid Cascade Neural Network Genetic Algorithm Space–Time Forecasting. *Symmetry* **2021**, *13*, 1158. https://doi.org/10.3390/sym13071158

Academic Editor: José Carlos R. Alcantud

Received: 7 May 2021
Accepted: 24 June 2021
Published: 28 June 2021

1. Introduction

Pollution describes the appearance and retention of the regular circulation of material, fine particles, biomaterial, and energy, or a deterioration technique or atmospheric change, which also has or may have significantly negative effects on human beings or the natural environment. Air pollutants are exhaust gases, particulate matter compounds, solid particulate matter, and other substances that emanate into the air, threatening the health of the community and damaging the environment. Air pollutants can be classified into smog and soot, pollution from contaminated air, greenhouse gas emissions, pollen, and mold.

PM refers to particulate matter, also known as particulate emissions. PM comprises aggregated rigid particles and atmospheric fluid droplets. Some are large enough or visible enough to be seen with the naked eye, and others are so small that they can only be seen with electron microscopes. PM_{10} and $PM_{2.5}$ are some classes of these particulate pollutants [1–4]. Let us consider a hair: the mean diameter of a single human hair is approximately 70 micrometers. This is roughly 28 times the diameter of $PM_{2.5}$. The diameter of particulate matter in PM_{10} is 10 micrometers or below. Similarly, $PM_{2.5}$ is normally particles of diameter 2.5 micrometers or below. Both PM_{10} and $PM_{2.5}$ are inhalable. We can thus imagine how tiny 2.5 and 10 micrometers are.

PM can be made up of various chemicals, including sulfur dioxide (SO_2) and nitrogen oxides, originating in PM (NOx) [5–7]. All this can be found as a product of building materials, farms, explosions, power stations, industry, and vehicles. PM is seriously damaging, as described above, as it may be opaque and small enough to be inhaled into the lungs or even into the circulation. Therefore, PM contamination affects the cardiovascular system and can cause fatal illnesses such as cardiovascular diseases, erratic heartbeat, and worsening asthma [8–10].

The estimation of future air pollution is an important task because it can be used to manage risk. The Artificial Neural Network (ANN) is the most frequently used among many data-driven applications and is a modern method and an effective paradigm for predicting and forecasting variables in the management of contamination risk due to intrinsic contaminant source uncertainties using quality data [11,12]. ANNs were inspired by the human brain's biochemical neural networks. McCulloch and Pitts in (1943) [13] initially developed a mathematically dependent model and referred to it as a threshold logic computing model for neural networks [14–19].

The neurons are important in the neural network's operating condition, they are very connected and share signals with one another, whether it is a neuron or node. Every layer consists of one or more simple elements called neurons. As the input data are transferred to the input layer, they bind with the weight and are nonlinearized by the activation function; the process of being sent to the next neuron is replicated before the final outcome is achieved. Each new neuron consists of one weight and one activation function [20,21]. The connectivity of neurons is handled by utilizing established inputs and outputs and is seen in an organized way in the ANN. The training phase is represented as a trial-and-error process to select the number of neurons [22,23]. The intensity of these interconnections is adapted to the known pattern using an error convergence technique. In this article, a cascade neural network procedure based on the genetic algorithm is developed for space–time forecasting data. This article is organized as follows: In Section 2, we review the training using the cascade neural network and employ a genetic algorithm. The performance is examined in Section 3 via simulation studies and analysis of four benchmark real datasets of air pollution data. Finally, Section 4 presents our conclusions.

2. Methods

2.1. Cascade Neural Network

The neural network's environment is uncertain. It is presumed that the teacher and the neural network are linked to an environmental testing vector, as an example. Because of the integrated experience, the teacher is capable of responding to this training vector in the algorithm. In fact, the appropriate outcome is the optimal response of the neural network. The key property of a neural network is the network's capacity to improve performance by learning from its experience [24]. Based on how well the neural networks operate, the networks are divided into supervised learning networks and unsupervised learning networks, otherwise termed teacher learning and teacher-free learning. Structurally, we may think of the teacher as having information about the environment, portrayed by a variety of samples of input and output [25,26].

The network parameters are managed according to the cumulative effect of the training vector, mostly with error signals. Meanwhile, each error signal should be specified as the gap between the requested response and the network's actual response. This modification is made step by step in order to rapidly mimic the teacher in the neural network, with the emulation in any mathematical context assumed to be ideal. This transfers awareness of the teacher's setting to the neural network as thoroughly as possible by preparation [27]. If this is achieved, then we can dispense with the teacher and encourage the neural network to deal entirely with the environment. Throughout the supervised NN model, input vectors and suitable target vectors are used to update the parameters; before a function can be approximated, input features should be tied to specific output vectors and the information that can be processed should be properly identified [28,29].

The most famous and typical algorithm for neural network training is the context of an error, the main principle of which is that an error in the hidden neurons is calculated by propagation of the error in the output layer neurons. The traditional backpropagation algorithm uses two input and learning processes. Vectors or patterns are displayed in the input layer in feedforward operation, and each neuron throughout the hidden layer is measured in the activation with one neuron net_j. The input vector dot product including neuron weight in the hidden layer is represented in Equation (1):

$$net_j = \sum_{i=1}^{N_i} x_i w_{ij}^{I-H} + b_{ij} = \sum_{i=0}^{N_i} x_i w_{ij}^{I-H} \tag{1}$$

where N_i is the input vector dimension, and i and j are neuron indices in the layer input and in the hidden layer, respectively. The weight value between the input vectors and neurons in the hidden layer is w_{ij}^{I-H}. The weight value of bias in the hidden layer is b_{ij} and is usually assumed to be $b_{ij} = w_{0j}^{I-0}$, $x_0 = 1$. By substituting net_j into activation function φ_1, θ_j is calculated. In the activation of a single neuron, each neuron in the output layer computes net_k, which is the dot product of θ_j and the neuron weight in the output layer, represented in Equations (2) and (3).

$$\theta_j = \varphi_1(net_j) \tag{2}$$

$$net_k = \sum_{j=1}^{N_H} \theta_j w_{jk}^{H-0} \tag{3}$$

In line with this, N_H is the number of neurons in the hidden layer and k is the index of a neuron in the output layer. The weight value of neurons between the hidden layer and output layer is described as w_{jk}^{H-0}. We can substitute net_k into activation function φ_2 to output y_k, represented in Equation (4).

$$y_k = \varphi_2(net_k) \tag{4}$$

$$y_k = \varphi_2\left(\sum_{j=1}^{N_H} \theta_j w_{jk}^{H-0}\right) \tag{5}$$

$$y_k = \varphi_2\left(\sum_{j=0}^{N_H} w_{jk}^{H-0}\left(\sum_{i=0}^{N_i} x_j w_{ij}^{I-H}\right)\right) \tag{6}$$

Regarding Equations (4) and (5), the entire collection of weights is updated to ensure that y_k is near the target output value of t_k by propagating the E_r error of the output layer neurons throughout the learning phase. Although a variety of output functions are available to evaluate the error, the squared error is commonly used, represented in Equation (7).

$$E_r = \sum_{k=1}^{N_0} (t_k - y_k)^2 \tag{7}$$

2.2. Genetic Algorithm

Biological variation and its basic processes were clarified by Darwin's (2002) evolutionary theory [30]. Natural selection is fundamental to what is often referred to as the macroscopic understanding of evolution. In an environment where only a finite number of humans will survive, and given the basic tendency of people to multiply, selection is necessary if people do not have an accelerated population [31,32]. This evolution favors people who bid more successfully for the given resources. In other words, they are better suited or adapted to the climate, recognized as global best survival [33].

Selection on the basis of competition is one of the two pillars of the mechanism of evolution. The other main influence comes as a function of phenotypical differences in the populations. The phenotype is an individual's physical and behavioral characteristics that assess their fitness in terms of their exposure to the surrounding environment. That individual represents a specific combination of environmental assessment phenotypic characteristics. These characteristics are inherited by the offspring of the individual if evaluated with favor; otherwise, the offspring is discarded. Charles Darwin's insight was that slight, spontaneous phenotype changes occur across ages [34–36].

New combinations of phenotype arise and are assessed by these mutations. That is the fundamental basis of the genetic algorithm: With a population of individuals, constraints on the environment lead to natural selection and survival of the population via roulette wheel selection, which results in an increase in the fitness of the population. A random collection of candidates can be generated [37]. Depending on this fitness, many of the best candidates are selected for the next generation using conjugation to seed the performance as an abstract fitness metric [38]. Cross-over and mutation give rise to a number of new offspring that fight for a position in the next generation on the basis of their fitness with old members of the population, before an organism of adequate efficiency is identified and until a previously determined computational threshold is exceeded [39,40]. In line with this, Algorithm 1 shows the scheme of the genetic algorithm. The scheme coincides with the generate-and-test algorithm type. The fitness function constitutes a heuristic estimate of an optimal solution, and the cross-over, mutation, and selection operators guide the search algorithm. The genetic algorithm has many characteristics that can support in the generating and testing of parents.

Algorithm 1. Scheme of the GA

```
INITIALIZE population and EVALUATE
while termination condition is not satisfied do
    SELECT parents
    CROSSOVER pairs of parents
    MUTATE the resulting offspring
    EVALUATE new candidates
    REPLACE individuals for the next generation
end while
```

2.3. Cascade Neural Network Genetic Algorithm

Backpropagation training algorithms based on other traditional optimization methods, such as the conjugating gradient and Newton process, have different variants. This same gradient descent approximation, the easiest and among the slowest, usually speeds up the conjugate gradient algorithm, as well as Newton's method [41,42]. We used genetic algorithms through this study. Each neuron weight between the hidden layer and the output layer should be updated, and the weight of the neurons here between the input and the hidden layer was adjusted [43]. The weight change between some of the hidden and output layers of the neuron is specified in Equation (8) with activation function $\varphi(x) = \frac{1}{x}$.

$$\frac{\partial E_r}{\partial w_{jk}^{H-0}} = \frac{\partial E_r}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{jk}^{H-0}}$$

$$\frac{\partial E_r}{\partial w_{jk}^{H-0}} = -(t_k - y_k) \cdot \varphi^2(net_k) \cdot \theta_j \tag{8}$$

The weight value of neurons was updated between the input and hidden layer as represented in Equation (9).

$$\frac{\partial E_r}{\partial w_{jk}^{I-H}} = \frac{\partial E_r}{\partial \theta_j} \cdot \frac{\partial \theta_j}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{jk}^{I-H}}$$

$$\frac{\partial E_r}{\partial w_{jk}^{I-H}} = \left[\frac{1}{2}\sum_{k=1}^{N_0}(t_k - y_k)^2\right] \cdot \frac{\partial \theta_j}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{jk}^{I-H}}$$

$$\frac{\partial E_r}{\partial w_{jk}^{I-H}} = \left[(t_k - y_k)^2 \cdot \frac{\partial y_k}{\partial \theta_j}\right] \cdot \frac{\partial \theta_j}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{jk}^{I-H}}$$

$$\frac{\partial E_r}{\partial w_{jk}^{I-H}} = \left[(t_k - y_k)^2 \cdot \frac{\partial y_k}{\partial net_k} \frac{\partial net_k}{\partial \theta_j}\right] \cdot \frac{\partial \theta_j}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{ij}^{I-H}}$$

$$\frac{\partial E_r}{\partial w_{jk}^{I-H}} = -\sum_{k=1}^{N_0}\left[(t_k - y_k) \cdot \varphi_2(net_k) \cdot w_{jk}^{H-0}\right] \cdot \varphi_1\left(net_j\right) \cdot x_i \tag{9}$$

With backpropagation, the input data are repeatedly presented to the neural network. With each presentation, the output of the neural network is compared to the desired output, and the error is computed. This error is then backpropagated through the neural network and used to adjust the weights such that the error decreases with each iteration; the neural network thus gets closer and closer to producing the desired output, represented in Equation (10).

$$w(h+1) = w(h) + \Delta w(h) \tag{10}$$

Algorithm 2 shows the function cascade neural network. However, the context backpropagation of each input datum is continuously shown to the neural network, with every representation comparing the output of the neural network to the requested output and computing the error; these errors provide context to the neural network and are used to update the weights to reduce its error for each iteration, as well as the genetic algorithm, allowing new generation of the neural network.

Algorithm 2. Function Cascade Neural Network

```
input $n_h$, $m$, $o$
set $k$ = 0
calculate Cascade Weighted
$k$ = 0;
for $i$ = 1:$nh$
    for $j$ = 1:$m$
        $k$ = $k$ + 1;
        $Wi1(i,j)$ = $W(k)$;
    end
end
calculate weighted input and output
for $i$ = 1:$o$
    for $j$ = 1:$m$
        $k$ = $k$ + 1;
        $Wi2(i,j)$ = $W(k)$;
    end
end
calculate weighted Bias Input
for $i$ = 1:$nh$
    $k$ = $k$ + 1;
    $Wbi(i,1)$ = $W(k)$;
end
calculate weighted output
for $i$ = 1:$o$
    for $j$ = 1:$nh$
        $k$ = $k$ + 1;
        $Wo(i,j)$ = $W(k)$;
    end
end
calculate weighted Bias Output
for $i$ = 1:$o$
    $k$ = $k$ + 1;
    $Wbo(i,1)$ = $W(k)$;
end
```

3. Simulation and Results

3.1. Construction of VAR-Cascade

There exist few guidelines for building a neural network model for time series. One of them considers time series as a nonlinear function of several past observations and random errors. Since air pollution data are known to be nonlinear time series data, we selected this method as a benchmark for forecasting. Equation (11) represents the time series models:

$$y_t = f[(z_{t-1}, z_{t-2}, \ldots, z_{t-m}),\ (e_{t-1}, e_{t-2}, \ldots,\ e_{t-n})] \tag{11}$$

where f is a nonlinear function determined by the neural network, $z_t = (1-B)^d\, y_t$, and d represents the order difference. Also, the residuals at time t are defined as e_t, and m and n are integers. Equation (12) shows that, initially, the VAR model is fitted in order to generate the residuals e_t. A neural network is then used to model the nonlinear and linear relations in excess and the original results [22,44,45].

$$z_t = w_0 + \sum_{j=1}^{Q} w_j \cdot g\left(w_{0j} + \sum_{i=1}^{p} w_{ij} \cdot z_{t-i} + \sum_{i=p+1}^{p+q} w_{ij} \cdot e_{t+p-i} \right) + \epsilon_t \tag{12}$$

Here, $w_{ij}(i = 0, 1, 2, \ldots, p + q, j = 1, 2, \ldots, Q)$ and $w_j(j = 0, 1, 2, 3, \ldots,\ Q)$ are connection weights and p, q, Q are integers that should be determined in the design process of the cascade neural network. The values of p and q are determined by the underlying properties of the data. If the data are just nonlinear, they only consist of nonlinear structures; then, q can be 0 since the Box–Jenkins method is a linear model that cannot simulate nonlinear interaction. Suboptimal methods may be used in a hybrid model, but suboptimality does not change the functional characteristics of the hybrid approach [17,46–48].

The interpretation of time series requires quantification of the vector dynamic response with time shifts. The main feature of this method is to forecast potential values using recent qualities of a variable, often referred to as lagged values [49]. Commonly, the latest values influence the estimation of a potential value most strongly [50,51]. A single scalar variable is frequently expressed in series data evaluation of a self-regression where future values are estimated based on the weighted total of pre-set lagged values. This variable relies on its own previous values as well as the previous values for many other variables in the much more specific multivariate case [52–54].

3.2. Study Area

The study areas were Taipei, Hsinchu, Taichung, and Kaohsiung city, with pollution data consisting of nitrogen oxide (NOx), atmospheric $PM_{2.5}$, atmospheric PM_{10}, and sulfur dioxide (SO_2) levels. Furthermore, the locations of these areas were as established by the Taiwan Environmental Protection Administration Executive Yuan. Table 1 shows statistical summaries of the amounts of air pollution at the four studied locations. The findings typically demonstrate that Taichung has higher concentrations of PM_{10}, $PM_{2.5}$, and NO_X, but in Kaohsiung, SO_2 is the greatest pollutant. Figure 1 shows an overview of the genetic algorithm's training and evaluation phases. Because each type of air pollutant has a different distribution, we trained the same models for each dataset using the same model architecture.

Table 1. Descriptive statistics.

Pollution	Location	N	Mean	SE Mean	StDev	Variance	Minimum	Q_1	Median	Q_3	Maximum	Range
PM_{10}	TAICHUNG	3632	50.642	0.419	24.949	622.476	5	32	45.5	65	173	168
	TAIPEI	3632	21.244	0.208	12.412	154.052	1	12	19	27	100	99
	HSINCHU	3632	22.46	0.219	13.135	172.521	1	13	19	29	103	102
	KAOHSIUNG	3632	31.719	0.31	18.477	341.414	1	17	29	44	123	122
SO_2	TAICHUNG	3632	2.8706	0.017	1.0208	1.042	0	2.2	2.7	3.4	9.3	9.3
	TAIPEI	3632	2.9835	0.0263	1.5742	2.4781	0.4	1.9	2.6	3.7	16.2	15.8
	HSINCHU	3632	2.6778	0.0186	1.1125	1.2377	0.1	1.9	2.5	3.2	13.2	13.1
	KAOHSIUNG	3632	5.3129	0.0515	3.0833	9.5066	0	3.3	4.5	6.4	33.8	33.8
$PM_{2.5}$	TAICHUNG	3632	26.623	0.264	15.66	245.244	1	15	23	35	106	105
	TAIPEI	3632	23.635	0.203	12.161	147.892	3.97	15.107	20.84	29.023	109.83	105.86
	HSINCHU	3632	18.179	0.13	7.763	60.265	0.63	13.04	16.34	21.102	76.72	76.09
	KAOHSIUNG	3632	23.854	0.163	9.747	95.014	5.27	16.49	21.495	29.697	68.96	63.69
NO_X	TAICHUNG	3632	22.944	0.171	10.26	105.269	4.35	15.23	20.57	28.63	81.43	77.08
	TAIPEI	3632	6.196	0.106	6.315	39.874	0.09	1.85	4.15	8.28	65.14	65.05
	HSINCHU	3632	3.2786	0.0546	3.2643	10.6559	0	1.52	2.3	3.79	45.65	45.65

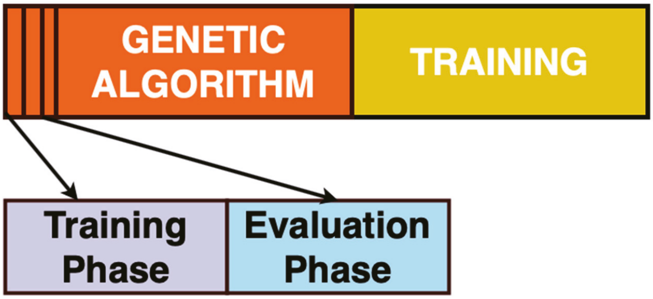


Figure 1. An overview of the genetic algorithm training and evaluation phases.

Samples for training were split in two, and alternating training and assessment were done in the first half of the samples. After this part was complete, the other half was used for forest training. Again, the first half was divided into smaller sections called stages. We perform simulations for ratios of 90:10, 80:20, 70:30, 60:40, and 50:50. In the training process, the training samples from the stepper were conditioned for all chromosomes, including new chromosomes of the previous level. Before the formation of the new chromosomes, all forests were educated in parallel. After all forests were qualified in the training part, genetic operators were used in the assessment part to calculate fitness values to operate in the genetic pool. This algorithm altered the substitute operator location to first and functioned only when a new chromosome was generated at the previous point.

3.3. Air Pollution Forecasting Using VAR-Cascade-GA

Poor air quality in Taiwan has mostly been identified as being a result of household burning, largely the source of greenhouse gas emissions. Taiwan's geography was observed to be a primary contributor towards its environmental problems, resulting in poor absorption and pollutant locking. Taipei, Taiwan's capital and most populous city, is surrounded by mountains, and advanced manufacturing offices all along the western and northern coastlines of Taiwan were also built near mountain ranges. In Section 3, we already discussed the construction step and simulation studies. Furthermore, during the construction stage of input, we employed the VAR pollution space–time dataset including Taichung (Y_1), Taipei (Y_2), Hsinchu (Y_3), and Kaohsiung (Y_4) in Taiwan.

Figure 2 shows that five hidden layers were used to create the model, and the ratio used was calculated by assessing the error values of the testing results shown in Table 2. During training and testing, $PM_{2.5}$ is represented in Figure 3, PM_{10} is represented in Figure 4, NO_X is represented in Figure 5, and SO_2 is represented in Figure 6. In this context, the cascade neural network genetic algorithm model can be used to study nonlinear and nonstationary data on air pollution. The metrics used to evaluate the test set's result were the root-mean-squared error (RMSE), mean absolute error (MAE), and symmetry mean absolute percentage (sMAPE) between the actual air pollution values and the predicted values. These are metrics that are commonly used in regression problems like our air pollution prediction. If all the metric values are smaller, then the model's performance is better [25].

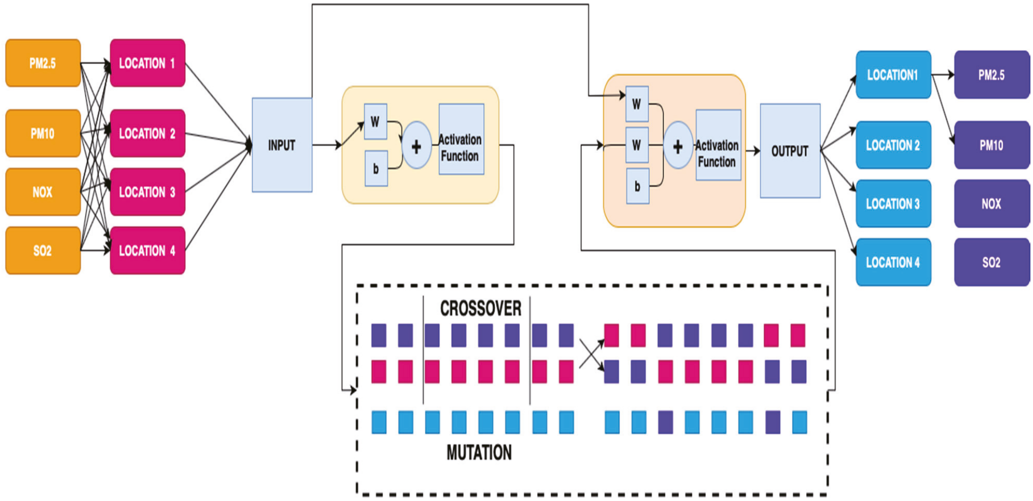


Figure 2. Space–time Cascade Neural Network with genetic algorithm, adopted by [22,36].

Table 2. Model comparison based on pollution.

Pollution	Portion	Training			Testing			Average			Elapsed Time
		RMSE	MAE	SMAPE	RMSE	MAE	SMAPE	RMSE	MAE	SMAPE	
$PM_{2.5}$	50:50	9.84	6.89	3.91	7.89	5.70	3.83	8.87	6.30	3.87	76.42
PM_{10}		13.53	9.57	3.84	11.37	8.05	3.86	12.45	8.81	3.85	71.86
NO_X		5.53	3.55	6.94	4.37	2.73	6.95	4.95	3.14	6.95	74.23
SO_2		1.72	1.18	3.60	1.15	0.80	3.61	1.44	0.99	3.61	75.51
$PM_{2.5}$	60:40	9.86	6.88	3.89	7.41	5.42	3.57	8.63	6.15	3.73	76.56
PM_{10}		13.49	9.54	3.76	10.67	7.57	3.56	12.08	8.55	3.66	75.24
NO_X		6.48	3.48	6.93	4.10	2.65	6.24	5.29	3.06	6.59	73.15
SO_2		1.65	1.13	3.48	1.07	0.80	3.43	1.36	0.97	3.46	76.13
$PM_{2.5}$	70:30	9.43	6.62	3.83	7.13	5.27	3.50	8.28	5.95	3.67	71.13
PM_{10}		13.30	9.29	3.74	10.09	7.13	3.74	11.70	8.21	3.74	77.87
NO_X		5.31	3.36	6.98	4.01	2.54	7.12	4.66	2.95	7.05	74.41
SO_2		1.60	1.09	3.36	1.00	0.77	3.41	1.30	0.93	3.38	80.90
$PM_{2.5}$	80:20	9.25	6.46	3.81	6.83	4.99	3.50	8.04	5.73	3.65	74.25
PM_{10}		13.10	9.07	3.74	9.19	6.56	4.13	11.14	7.82	3.93	74.25
NO_X		5.25	3.29	7.11	3.79	2.49	6.68	4.52	2.89	6.90	72.37
SO_2		1.54	1.05	3.42	0.92	0.70	3.36	1.23	0.88	3.39	80.24
$PM_{2.5}$	90:10 *	9.03	6.34	3.77	6.78	4.94	3.47	7.90	5.64	3.62	83.83
PM_{10}		12.77	8.85	3.77	8.02	5.93	4.09	10.40	7.39	3.93	75.00
NO_X		5.11	3.20	6.90	3.70	2.43	6.53	4.40	2.81	6.72	80.47
SO_2		1.48	1.10	3.37	0.80	0.62	2.77	1.14	0.86	3.07	71.71

Noted: Best simulation with low error (*) and yellow highlight represent the lowest value of each information pollution, accuracy measurement, and elapsed time.

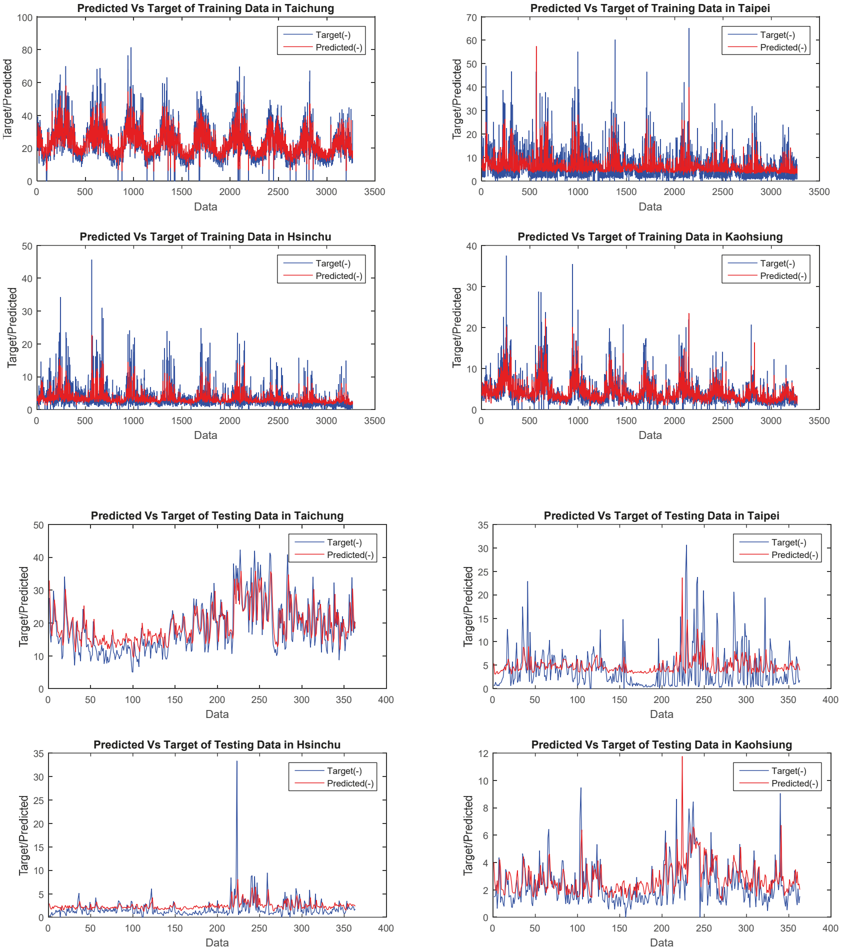


Figure 3. $PM_{2.5}$ data training of the CFNN using a genetic algorithm and backpropagation.

In the results, the cascade neural network genetic algorithm with ratio 90:10 provided lower RMSE, MAE, and sMAPE values for all variables. The optimum number of hidden neurons showing good performance in the test and validation results could then be selected. Using this model, a prediction of future air pollution was performed. In this study area, the air pollution levels in these four cities in Taiwan influence each other. However, the accuracy of prediction was not improved when we set the training and testing ratio to 80:20, 70:30, 60:40, or 50:50 in the same section. There are several training algorithms, such as backpropagation, Conjugate Gradient Powell–Beale (CGB), Broyden–Fletcher–Goldfarb (BFG), Levenberg–Marquardt (LM), and Scaled Conjugate Gradient (SCG). The rate of change in the error with respect to the connection weight including the error gradient is used as a path for training.

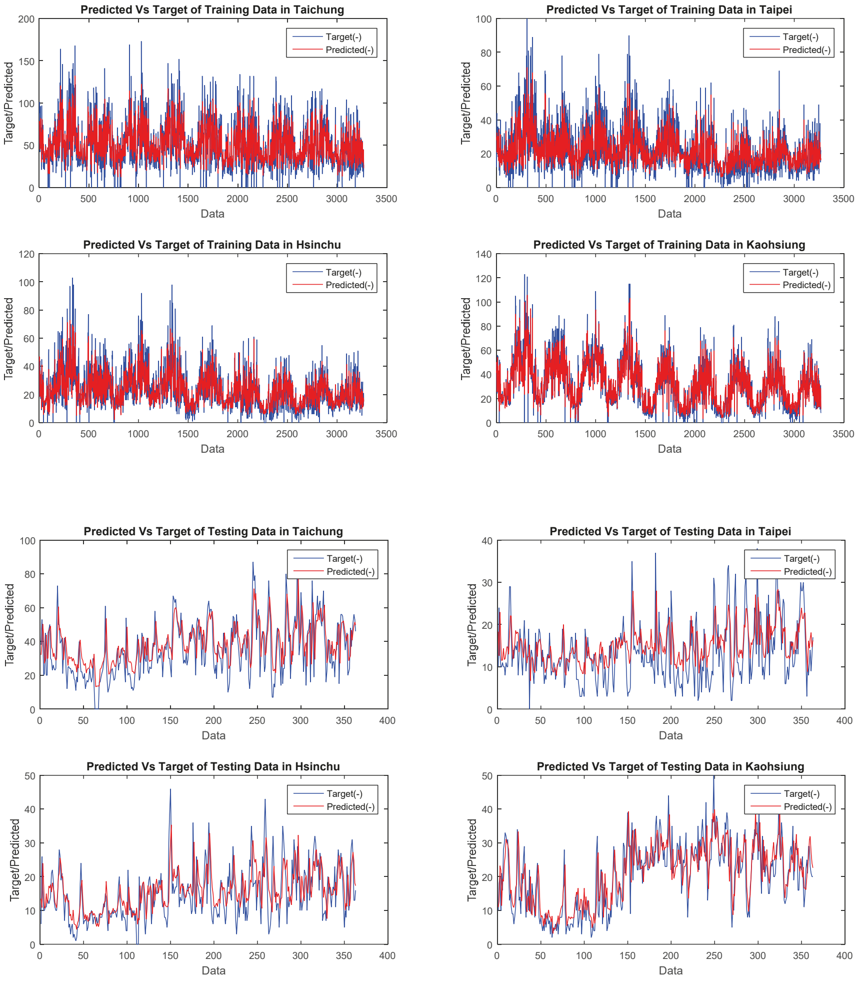


Figure 4. PM$_{10}$ data training of the CFNN using a genetic algorithm and backpropagation.

In order to measure the stage size to optimize performance, we used backpropagation and conducted a search along the conjugate or orthogonal path. Appropriately, we proved that this was the easiest way to train moderate feedback networks. That being said, some matrix multiplication is included in the processing for such issues as air pollution over time. The network is very wide in this research, so using backpropagation is a good way. When overfitting occurs, the transferability of the potential is significantly decreased. To suppress overfitting, methods such as so-called regularization are often used. L1 (L2) regularization adds the sum of the absolute (square) values of weights to the loss function, as in Equations (13) and (14), where Γ is the loss function and w^i_{jk} indicates the weights in the network. In addition, α is the scaling factor for the summation. N_H denotes the number of layers and N_i denotes the number of nodes in the ith layer.

$$\Gamma_{L1} = \Gamma + \alpha \sum_{i=1}^{N_H - 1} \sum_{j=1}^{N_i} \sum_{k=1}^{N_{i+1}} \left| w^i_{jk} \right| \tag{13}$$

$$\Gamma_{L2} = \Gamma + \alpha \sum_{i=1}^{N_H-1} \sum_{j=1}^{N_i} \sum_{k=1}^{N_{i+1}} \left(w_{jk}^{i}\right)^2 \tag{14}$$

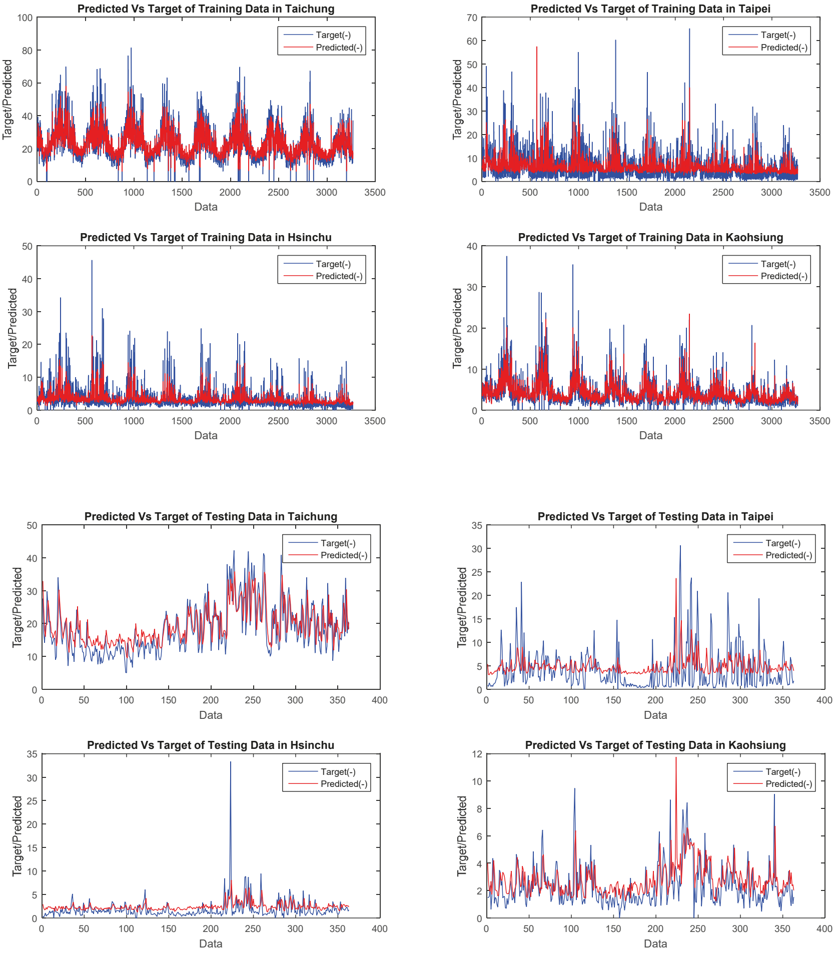


Figure 5. NO_x data training of the CFNN using a genetic algorithm and backpropagation.

3.4. Does the Activation Function Provide High Accuracy and Speed Up the Time Lapse?

Linear regression models work well throughout short-term predictions based on daily or weekly measurements in time series forecasting, but they cannot tackle nonlinearity in showing variables properly, not even for long-term predictions from seasonal or annual data series. Various machine learning methodologies have been introduced and used to simulate problems and provide predictions in environmental research, as machine efficiency has been evolving rapidly in the last decade. Despite its prominence and outstanding data accuracy, critical issues in the Artificial Neural Network are its propensity to overfit training data and inconsistency for short histories of training data. Several strategies for more effective and efficient preparation of NNs have been recommended. However, these are not simple and also have markedly poor accuracy.

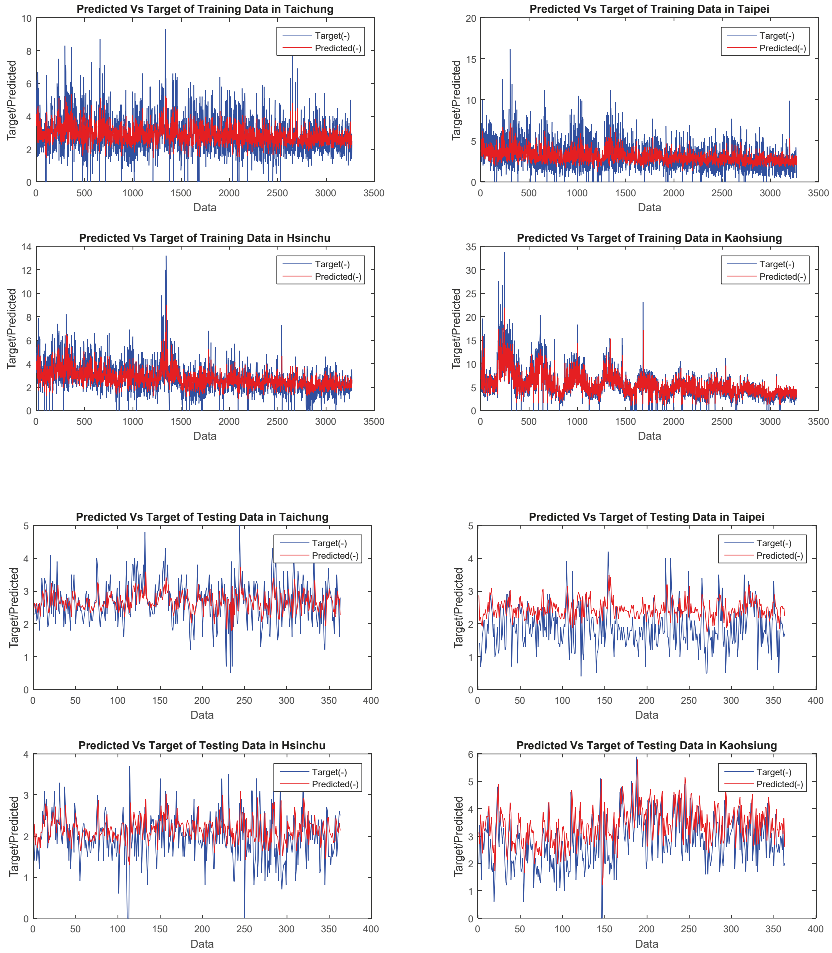


Figure 6. SO_2 data training of the CFNN using a genetic algorithm and backpropagation.

After the training and testing comparisons already discussed in Section 3.3, we considered proving the performance of the hybrid cascade neural network genetic algorithm when using other activation functions. Computational capabilities are increasing in the era of big data, high-performance computing, parallel processing, and cloud computing. In line with this, we address whether the activation function can improve accuracy and speed up the time lapse. Throughout the last decades, the machine learning domain, a branch of artificial intelligence, has gained popularity, and researchers in the area have led it to expand through various areas of human life. Machine learning is a part of research that employs statistics and computer science concepts to develop mathematical models used to execute large tasks such as estimation and inference [55]. These frameworks are collections of mathematical interactions between the system's inputs and outputs. A learning process entails predicting the model parameters so that the task can be executed effectively. To improve accuracy, researchers have conducted simulated comparisons using various activation functions. The most popular activation functions are SoftMax, tanh, ReLU, Leaky ReLU, sigmoid, and logsig [56–59].

As asserted, the activation function can be defined and applied to an ANN to assist the network in understanding various systems in data. Although contrasted to a neuron-based design seen in human brains, an activation function is essentially responsible for determining what neuron to trigger immediately [60]. Inside an ANN, the activation function is doing the same thing. All of this receives a prior nerve cell output signal and transforms it into a format which can be used as feedback to yet another cell. In this simulation, we used logsig in Equation (15), radbas in Equation (16), SoftMax in Equation (17), and tribas in Equation (18).

$$z_j = \frac{1}{1 + \exp(-X_j)} \tag{15}$$

$$z(x) = \sum_{i=1}^{N} w_i \varphi(\|x - x_i\|) \tag{16}$$

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{17}$$

$$tri(x) = \Lambda(x) \overset{\text{def}}{=} \max(1 - |x|, 0) \tag{18}$$

Table 3 shows that the best activation function for PM_{10} was logsig, that for $PM_{2.5}$ was SoftMax, that for NO_x was radbas, and that for SO_2 was tribas. The SoftMax activation function provided a shorter time lapse than other activation functions.

Table 3. Combining activation functions with the Cascade Neural Network.

Pollution	Activation Function	Training			Testing			Average			Elapsed Time
		RMSE	MAE	SMAPE	RMSE	MAE	SMAPE	RMSE	MAE	SMAPE	
$PM_{2.5}$	logsig	9.02	6.30	3.77	6.77	4.94	3.54	7.90	5.62	3.66	78.03
PM_{10}		12.79	8.86	3.71	8.12	6.02	4.05	10.46	7.44	3.88	75.48
NO_X		5.04	3.15	6.84	3.79	2.43	6.62	4.42	2.79	6.73	72.43
SO_2		1.48	1.01	3.35	0.82	0.63	3.00	1.15	0.82	3.18	76.59
$PM_{2.5}$	radbas	9.06	6.37	3.80	6.80	5.00	3.51	7.93	5.69	3.66	80.97
PM_{10}		12.79	8.85	3.79	8.11	6.03	4.19	10.45	7.44	3.99	74.77
NO_X		5.09	3.16	6.83	3.75	2.40	6.19	4.42	2.78	6.51	82.35
SO_2		1.49	1.01	3.36	0.83	0.64	2.93	1.16	0.83	3.15	73.33
$PM_{2.5}$	SoftMax	9.07	6.35	3.78	6.70	4.91	3.46	7.89	5.63	3.62	75.47
PM_{10}		12.81	8.90	3.75	8.13	6.04	4.24	10.47	7.47	3.99	74.70
NO_X		5.11	3.19	7.12	3.64	2.35	6.23	4.38	2.77	6.67	72.11
SO_2		1.48	1.01	3.42	0.84	0.66	3.12	1.16	0.84	3.27	77.53
$PM_{2.5}$	tribas	9.03	6.34	3.80	6.81	4.97	3.52	7.92	5.66	3.66	93.20
PM_{10}		12.81	8.90	3.75	8.13	6.04	4.24	10.47	7.47	3.99	74.70
NO_X		5.13	3.21	6.98	3.68	2.37	6.53	4.41	2.79	6.75	72.29
SO_2		1.49	1.01	3.36	0.84	0.65	2.94	1.16	0.83	3.15	80.35

Noted: Yellow highlight represent the lowest value of each information pollution, accuracy measurement, and elapsed time.

The cascade feed forward neural network model differs only when determining the input variables. During the simulation, we constructed the input by vector autoregression. Then, we considered the input as the lag variable of each predicted variable, in this case, the air pollution data at the four locations of Taichung, Taipei, Hsinchu, and Kaohsiung. Then, in the CFNN model for the four locations, neurons were compiled in the layer and the signal from the input to the first input layer, then to the second layer (hidden layer), and finally to the output layer. The general equation for forecasting pollution data in the four locations, represented in Equation (19), was used for prediction purposes in these study areas. Meanwhile, Equation (20) shows four input neurons Y_{t-1} (lag 1) and five neurons in hidden layer of Z_t. To perform the forecasting, we used Equation (21) for NO_x with the radial basis activation function, Equation (22) for $PM_{2.5}$ with the SoftMax activation function, Equation (20) for PM_{10} with the logsig activation function, and Equation (23) for SO_2 with the tribas activation function. We provide the results of forecasting in Figure 7 for the next 30 steps. The results show Taichung constantly leading with the highest pollutant score compared to other cities in Taiwan.

$$\begin{aligned} \hat{Y}_t &= \psi_2\left\{ [w_{b0}\ w_0\ w_{i2}] \begin{bmatrix} 1 \\ Z_t \\ Y_{t-1} \end{bmatrix} \right\} \\ Z_t &= \psi_2\left(\begin{bmatrix} w_{bi} & w_{i1} \end{bmatrix} \begin{bmatrix} 1 \\ Y_{t-1} \end{bmatrix} \right) \end{aligned} \tag{19}$$

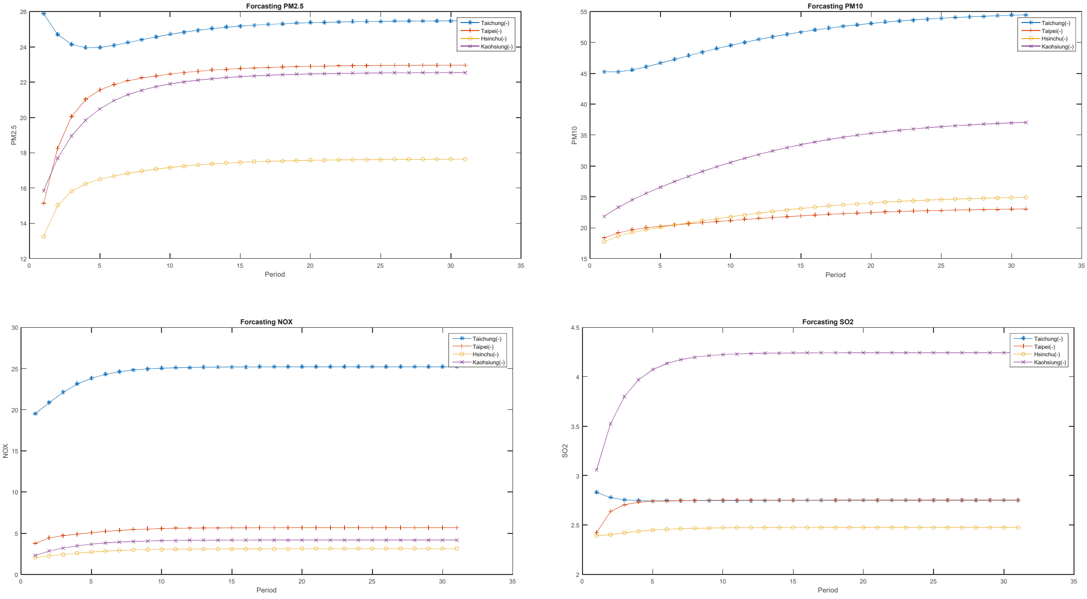


Figure 7. Forecasting all pollution datasets using the CFNN with a genetic algorithm and backpropagation.

Cascade Neural Network Genetic Algorithm for NO_x using the radial basis activation function:

$$\hat{Y}_t = \begin{bmatrix} -0.1244 & -0.0992 & 0.01054 & -7.4544 & 0.2238 & -6.9588 & 0.5603 & 0.0444 & -0.0689 & 0.0770 \\ -0.2483 & 0.0359 & -0.0665 & -6.6298 & 1.0709 & -1.7180 & 0.0655 & 0.2562 & 0.3422 & 0.0572 \\ -0.4328 & 0.0253 & -0.0226 & 4.9796 & 0.8715 & 5.0640 & 0.1253 & 0.0189 & 0.3570 & 0.0742 \\ -0.2772 & 0.0015 & 0.1201 & -4.45319 & -0.1428 & 4.9089 & 0.1066 & 0.0465 & -0.0148 & 0.5544 \end{bmatrix} \begin{bmatrix} 1 \\ Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{3,t-1} \\ Y_{4,t-1} \end{bmatrix},$$

$$Z_t = radbas\begin{bmatrix} 5.0776 & 6.4191 & 0.0893 & 0.7930 & -0.2450 \\ 0.8287 & 4.8674 & -2.4160 & -3.1740 & -0.7644 \\ -6.9722 & -4.2852 & 9.1216 & -3.0205 & 8.5264 \\ -9.6954 & -12.9182 & 5.3748 & 18.2621 & -5.5156 \\ 7.7948 & 2.5076 & 2.6535 & -9.1934 & -9.4768 \end{bmatrix}\begin{bmatrix} 1 \\ Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{3,t-1} \\ Y_{4,t-1} \end{bmatrix} \tag{20}$$

Cascade Neural Network Genetic Algorithm for $PM_{2.5}$ using the SoftMax activation function:

$$\begin{bmatrix} 0.9653 & -1.0657 & 3.2999 & -2.4456 & -0.7936 & -1.1990 & 0.7359 & 0.0491 & -0.0690 & 0.1124 \\ 1.2965 & -1.5045 & 2.3489 & 2.0493 & -1.9046 & -1.6001 & 0.0225 & 0.2826 & 0.3253 & 0.0500 \\ 6.0285 & -6.1880 & 0.3716 & 2.3998 & -6.8009 & -6.3181 & 0.1929 & -0.0279 & 0.4639 & 0.1170 \\ 0.9319 & -0.9456 & 10.3730 & 3.4200 & -1.1032 & -0.9099 & 0.0890 & 0.0227 & 0.0463 & 0.7191 \end{bmatrix}\begin{bmatrix} 1 \\ Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{3,t-1} \\ Y_{4,t-1} \end{bmatrix},$$

$$Z_t = softmax\begin{bmatrix} 6.2714 & -7.3511 & -7.8139 & -1.9507 & 4.3787 \\ -5.6864 & -7.7599 & -8.3507 & -2.9367 & -2.3266 \\ -6.4463 & -6.6875 & 5.0926 & 1.5264 & -1.1062 \\ -5.0961 & -3.3109 & 5.3424 & 6.5753 & -3.4388 \\ 3.6740 & -2.6485 & -6.2284 & -6.7208 & 4.4362 \end{bmatrix}\begin{bmatrix} 1 \\ Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{3,t-1} \\ Y_{4,t-1} \end{bmatrix} \tag{21}$$

Cascade Neural Network Genetic Algorithm for PM_{10} using the logsig activation function:

$$\hat{\boldsymbol{Y}}_t = \begin{bmatrix} 4.7332 & -0.0171 & -4.7892 & -0.0294 & -0.0499 & -0.0380 & 0.5673 & 0.0698 & 0.0402 & 0.0720 \\ -4.7021 & 0.0480 & 4.5307 & -0.0609 & 0.0313 & -0.1244 & 0.1252 & 0.3677 & 0.1878 & 0.0614 \\ 2.2770 & 0.0263 & -2.4010 & -0.0006 & -0.0102 & 0.0014 & 0.2351 & -0.0648 & 0.5960 & 0.0696 \\ 1.5241 & -0.0534 & -1.4885 & 0.0533 & -0.1240 & 0.1097 & 0.1118 & 0.1017 & 0.1786 & 0.5507 \end{bmatrix}\begin{bmatrix} 1 \\ Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{3,t-1} \\ Y_{4,t-1} \end{bmatrix},$$

$$Z_t = logsig\begin{bmatrix} -7.6484 & -5.6012 & 5.5769 & 0.2050 & -7.1292 \\ 9.0472 & -2.0909 & -7.4709 & 4.9145 & -5.3312 \\ -4.1385 & -9.9840 & -3.4291 & 6.0077 & 2.4277 \\ -1.5068 & -4.0147 & 3.5676 & 5.6306 & -9.0051 \\ -5.6488 & 5.5027 & 6.1290 & 7.2003 & 6.3822 \end{bmatrix}\begin{bmatrix} 1 \\ Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{3,t-1} \\ Y_{4,t-1} \end{bmatrix} \tag{22}$$

Cascade Neural Network Genetic Algorithm for SO_{10} using the tribas activation function:

$$\hat{Y}_t = \begin{bmatrix} -0.1116 & 5.1674 & -0.0036 & -0.0166 & -3.4727 & -0.2061 & 0.4570 & 0.0222 & 0.0743 & 0.0524 \\ -0.2789 & -1.0341 & -0.0087 & -0.0114 & 7.8267 & 0.1241 & 0.0817 & 0.3550 & 0.1215 & 0.0309 \\ -0.8104 & -7.8058 & -0.0085 & 0.0234 & 5.2240 & 0.0292 & 0.1364 & 0.0647 & 0.4179 & 0.1027 \\ -0.1772 & 6.5887 & -0.0242 & -0.0490 & 6.7444 & 0.0271 & 0.0246 & 0.0067 & 0.0693 & 0.6512 \end{bmatrix} \begin{bmatrix} 1 \\ Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \\ Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{3,t-1} \\ Y_{4,t-1} \end{bmatrix},$$

$$Z_t = tribas \begin{bmatrix} 7.3483 & -2.1911 & -8.1641 & 0.4300 & -3.6214 \\ -1.5728 & -2.5029 & -10.1020 & 0.5066 & 7.0828 \\ -5.3810 & -9.9016 & 7.7806 & -2.1473 & 5.0866 \\ -1.1201 & 10.2142 & 6.6401 & 6.7203 & 4.0480 \\ -2.9485 & 8.1185 & 7.2550 & 8.2102 & -8.3387 \end{bmatrix} \begin{bmatrix} 1 \\ Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{3,t-1} \\ Y_{4,t-1} \end{bmatrix} \tag{23}$$

4. Conclusions

In this paper, we first presented a full review of a cascade neural network with a genetic algorithm as applied to space–time forecasting. Experimental results on an air pollution dataset showed that our hybrid methods provide high accuracy as proved by the RMSE, MAE, and sMAPE values. Attributable to its rapid urbanization and industrialization over the last decades, Taiwan faces serious environmental issues, including air pollution. In order to resolve air quality issues, the government has taken several countermeasures. The attempt to eliminate SO_2 and overall suspended particulate matter was very effective when ever-increasing cars threatened city atmospheres with NOx and particulates. A space–time air pollution analysis over the last 10 years using the monitoring data clearly showed that with urban planning and countermeasure policies, air quality has improved. The analysis should be used to make future policy decisions. Air pollution temporal features were examined herein for Taiwan. The pattern from pollutants to particulates differs in air quality for each location. In a nutshell, the PM, SO_2, and NO_x levels have drastically increased. Future research should examine using VAR-SARIMA, VAR-ARCH, and other traditional time series as input.

Author Contributions: Conceptualization, R.E.C., H.Y.; methodology, R.E.C., H.Y.; software, R.E.C., H.Y.; validation, R.E.C., H.Y.; formal analysis, R.E.C., H.Y.; investigation, R.E.C., H.Y.; resources, R.E.C., H.Y.; Writing—original draft, R.E.C., H.Y.; writing—review and editing, R.E.C., H.Y.; visualization, R.E.C., H.Y.; supervision, R.E.C., H.Y., R.-C.C., M.B.; project administration, R.E.C., H.Y., R.-C.C., N.E.G., B.D.S., T.T., M.B., P.U.G., B.P.; funding acquisition, R.E.C., R.-C.C., T.T., M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research fully supported by Faculty of Business and Economics, University of Indonesia. This research is part of Ministry of Science and Technology, Taiwan [MOST-109-2622-E-324-004]. This research is part of Chaoyang University of Technology and the Higher Education Sprout Project, Ministry of Education (MOE), Taiwan, under the project name: "The R&D and the cultivation of talent for health-enhancement products". This research is fully supported by the Directorate General of Research and Community Service, the Ministry of Research, and Technology/National Agency for Research and Innovation of the Republic of Indonesia through World-Class Research Program 2021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Supplementary code to this article can be found online at https://github.com/Rezzy94/cascadenn (accessed on 1 May 2021). The copyright of this programming was officially registered on May 31, 2021, by the Directorate General of Intellectual Property—Ministry of Law and Human Rights, the Republic of Indonesia, with the registration number [EC00202125523] and [000252427], valid for 50 (fifty) years from the first announcement of the work. This copyright registration letter or related rights products are in accordance with article 72 of law number 28 of 2014 concerning copyright.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

VAR: vector autoregression, GA: genetic algorithm, FFNN: feedforward neural network, MAE: mean absolute error, sMAPE: symmetric mean absolute percentage error, RMSE: root-mean-square error, MAE: mean absolute error, $PM_{2.5}$: fine particulate matter with a diameter of 2.5 µm, PM_{10}: fine particulate matter 10 micrometers or less in diameter, SO_2: sulfur dioxide, NO_x: nitrogen dioxide, BP: backpropagation.

References

1. Querol, X.; Alastuey, A.; Ruiz, C.R.; Artiñano, B.; Hansson, H.C.; Harrison, R.M.; Buringh, E.; Ten Brink, H.M.; Lutz, M.; Bruckmann, P.; et al. Speciation and origin of PM10 and PM2.5 in selected European cities. *Atmos. Environ.* **2004**, *38*, 6547–6555. [CrossRef]
2. Fan, J.; Wu, L.; Ma, X.; Zhou, H.; Zhang, F. Hybrid support vector machines with heuristic algorithms for prediction of daily diffuse solar radiation in air-polluted regions. *Renew. Energy* **2020**, *145*, 2034–2045. [CrossRef]
3. Masseran, N.; Safari, M.A.M. Intensity–duration–frequency approach for risk assessment of air pollution events. *J. Environ. Manag.* **2020**, *264*, 110429. [CrossRef] [PubMed]
4. Masseran, N.; Safari, M.A.M. Modeling the transition behaviors of PM 10 pollution index. *Environ. Monit. Assess.* **2020**, *192*, 441. [CrossRef] [PubMed]
5. De Vito, S.; Piga, M.; Martinotto, L.; Di Francia, G. CO, NO_2 and NOx urban pollution monitoring with on-field calibrated electronic nose by automatic bayesian regularization. *Sens. Actuators B Chem.* **2009**, *143*, 182–191. [CrossRef]
6. Winarso, K.; Yasin, H. Modeling of air pollutants SO_2 elements using geographically weighted regression (GWR), geographically temporal weighted regression (GTWR) and mixed geographically temporalweighted regression (MGTWR). *ARPN J. Eng. Appl. Sci.* **2016**, *11*, 8080–8084.
7. Zhang, J.J.; Wei, Y.; Fang, Z. Ozone pollution: A major health hazard worldwide. *Front. Immunol.* **2019**, *10*, 2518. [CrossRef]
8. Bernstein, J.A.; Alexis, N.; Barnes, C.; Bernstein, I.L.; Bernstein, J.A.; Nel, A.; Peden, D.; Diaz-Sanchez, D.; Tarlo, S.M.; Williams, P.B. Health effects of air pollution. *J. Allergy Clin. Immunol.* **2004**, *114*, 1116–1123. [CrossRef]
9. Xing, Y.F.; Xu, Y.H.; Shi, M.H.; Lian, Y.X. The impact of PM2.5 on the human respiratory system. *J. Thorac. Dis.* **2016**, *8*, E69–E74.
10. Rossati, A. Global warming and its health impact. *Int. J. Occup. Environ. Med.* **2017**, *8*, 7–20. [CrossRef]
11. Suhartono, S.; Subanar, S. Development of model building procedures in wavelet neural networks for forecasting non-stationary time series. *Eur. J. Sci. Res.* **2009**, *34*, 416–427.
12. Suhermi, N.; Suhartono; Prastyo, D.D.; Ali, B. Roll motion prediction using a hybrid deep learning and ARIMA model. *Procedia Comput. Sci.* **2018**, *144*, 251–258. [CrossRef]
13. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
14. Chen, R.C.; Dewi, C.; Huang, S.W.; Caraka, R.E. Selecting critical features for data classification based on machine learning methods. *J. Big Data* **2020**, *7*, 52. [CrossRef]
15. Caraka, R.E.; Lee, Y.; Chen, R.C.; Toharudin, T. Using Hierarchical Likelihood towards Support Vector Machine: Theory and Its Application. *IEEE Access* **2020**, *8*, 194795–194807. [CrossRef]
16. Mueller, J.-A.; Lemke, F. Self-Organising Data Mining: An Intelligent Approach to Extract Knowledge from Data. 1999. Available online: https://www.knowledgeminer.eu/pdf/sodm.pdf (accessed on 6 May 2021).
17. De Gooijer, J.G.; Hyndman, R.J. 25 years of time series forecasting. *Int. J. Forecast.* **2006**, *22*, 443–473. [CrossRef]
18. Kaimian, H.; Li, Q.; Wu, C.; Qi, Y.; Mo, Y.; Chen, G.; Zhang, X.; Sachdeva, S. Evaluation of different machine learning approaches to forecasting PM2.5 mass concentrations. *Aerosol Air Qual. Res.* **2019**, *19*, 1400–1410. [CrossRef]
19. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48. [CrossRef]
20. Szandała, T. Review and comparison of commonly used activation functions for deep neural networks. *arXiv* **2020**. Available online: https://arxiv.org/abs/2010.09458 (accessed on 6 May 2021).
21. Sony, S.; Dunphy, K.; Sadhu, A.; Capretz, M. A systematic review of convolutional neural network-based structural condition assessment techniques. *Eng. Struct.* **2021**, *226*, 111347. [CrossRef]
22. Caraka, R.E.; Chen, R.C.; Yasin, H.; Pardamean, B.; Toharudin, T.; Wu, S.H. Prediction of Status Particulate Matter 2.5 using State Markov Chain Stochastic Process and HYBRID VAR-NN-PSO. *IEEE Access* **2019**, *7*, 161654–161665. [CrossRef]
23. Kuster, C.; Rezgui, Y.; Mourshed, M. Electrical load forecasting models: A critical systematic review. *Sustain. Cities Soc.* **2017**, *35*, 257–270. [CrossRef]

24. Cios, K.J.; Pedrycz, W.; Swiniarski, R.W.; Kurgan, L.A. *Data Mining: A Knowledge Discovery Approach*; Springer: Boston, MA, USA, 2007; ISBN 9780387333335.
25. Makridakis, S.; Spiliotis, E.; Assimakopoulos, V. The M4 Competition: 100,000 time series and 61 forecasting methods. *Int. J. Forecast.* **2020**, *36*, 54–74. [CrossRef]
26. Makridakis, S.G.; Wheelwright, S.C.; Hyndman, R.J. Forecasting: Methods and Applications. *J. Forecast.* **1998**, 1–656. [CrossRef]
27. Wong, K.W.; Wong, P.M.; Gedeon, T.D.; Fung, C.C. Rainfall prediction model using soft computing technique. *Soft Comput.* **2003**, *7*, 434–438. [CrossRef]
28. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
29. Mislan, M.; Haviluddin, H.; Hardwinarto, S.; Sumaryono, S.; Aipassa, M. Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggarong Station, East Kalimantan—Indonesia. *Procedia Comput. Sci.* **2015**, *59*, 142–151. [CrossRef]
30. Darwin, C. *The Correspondence of Charles Darwin: 1821–1860*; Cambridge University Press: Cambridge, UK, 2002.
31. Pfeiffer, J.R. Evolutionary theory. In *George Bernard Shaw in Context*; Cambridge University Press: Cambridge, UK, 2015; ISBN 9781107239081.
32. Wuketits, F.M. Charles darwin and modern moral philosophy. *Ludus Vitalis* **2009**, *17*, 395–404.
33. García-Martínez, C.; Rodriguez, F.J.; Lozano, M. Genetic algorithms. In *Handbook of Heuristics*; Springer: Cham, Switzerland, 2018; ISBN 9783319071244. Available online: https://www.springer.com/gp/book/9783319071237 (accessed on 6 May 2021).
34. Sivanandam, S.; Deepa, S. *Introduction to Genetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2008.
35. Gupta, J.N.D.; Sexton, R.S. Comparing backpropagation with a genetic algorithm for neural network training. *Omega* **1999**, *27*, 679–684. [CrossRef]
36. Caraka, R.E.; Chen, R.C.; Yasin, H.; Lee, Y.; Pardamean, B. Hybrid Vector Autoregression Feedforward Neural Network with Genetic Algorithm Model for Forecasting Space-Time Pollution Data. *Indones. J. Sci. Technol.* **2021**, *6*, 243–266.
37. Kubat, M.; Kubat, M. The Genetic Algorithm. In *An Introduction to Machine Learning*; Springer International Publishing: Cham, Switzerland, 2017.
38. Moscato, P.; Cotta, C. A Modern Introduction to Memetic Algorithms. In *Handbook of Metaheuristics*; Springer: Boston, MA, USA, 2010; pp. 141–183. Available online: https://link.springer.com/chapter/10.1007/978-1-4419-1665-5_6 (accessed on 6 May 2021).
39. Makridakis, S.; Wheelwright, S.C. Forecasting Methods for Management. *Oper. Res. Q.* **1974**, *25*, 648–649. [CrossRef]
40. Makridakis, S. A Survey of Time Series. *Int. Stat. Rev. Rev. Int. Stat.* **1976**, *44*, 29. [CrossRef]
41. Warsito, B.; Santoso, R.; Suparti; Yasin, H. Cascade Forward Neural Network for Time Series Prediction. *J. Phys. Conf. Ser.* **2018**, *1025*, 012097. [CrossRef]
42. Schetinin, V. A learning algorithm for evolving cascade neural networks. *Neural Process. Lett.* **2003**, *17*, 21–31. [CrossRef]
43. Ding, S.; Zhao, H.; Zhang, Y.; Xu, X.; Nie, R. Extreme learning machine: Algorithm, theory and applications. *Artif. Intell. Rev.* **2015**, *44*, 103–115. [CrossRef]
44. Suhartono; Prastyo, D.D.; Kuswanto, H.; Lee, M.H. Comparison between VAR, GSTAR, FFNN-VAR and FFNN-GSTAR Models for Forecasting Oil Production Methods. *Mat. Malays. J. Ind. Appl. Math.* **2018**, *34*, 103–111.
45. Prastyo, D.D.; Nabila, F.S.; Lee, M.H.S.; Suhermi, N.; Fam, S.F. VAR and GSTAR-based feature selection in support vector regression for multivariate spatio-temporal forecasting. In *Communications in Computer and Information Science*; Springer: Singapore, 2018; pp. 46–57.
46. Zhang, P.G. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **2003**, *50*, 159–175. [CrossRef]
47. Geurts, M.; Box, G.E.P.; Jenkins, G.M. Time Series Analysis: Forecasting and Control. *J. Mark. Res.* **2006**. [CrossRef]
48. McLeod, A.I.; Yu, H.; Mahdi, E. Time Series Analysis with R. *Handb. Stat.* **2011**, *30*, 661–672. [CrossRef]
49. Liao, W.T. Clustering of time series data—A survey. *Pattern Recognit.* **2005**, *38*, 1857–1874. [CrossRef]
50. Subba Rao, T. Time Series Analysis. *J. Time Ser. Anal.* **2010**, *31*, 139. [CrossRef]
51. Mudelsee, M. *Climate Time Series Analysis: Regression*; Springer: Dordrecht, The Netherlands, 2010; Volume 42, ISBN 978-90-481-9481-0.
52. Zhu, X.; Pan, R.; Li, G.; Liu, Y.; Wang, H. Network vector autoregression. *Ann. Stat.* **2017**, *45*, 1096–1123. [CrossRef]
53. Nourani, V.; Baghanam, A.H.; Adamowski, J.; Gebremichael, M. Using self-organizing maps and wavelet transforms for space-time pre-processing of satellite precipitation and runoff data in neural network based rainfall-runoff modeling. *J. Hydrol.* **2013**, *476*, 228–243. [CrossRef]
54. Ippoliti, L.; Valentini, P.; Gamerman, D. Space-time modelling of coupled spatiotemporal environmental variables. *J. R. Stat. Soc. Ser. C Appl. Stat.* **2012**. [CrossRef]
55. Sharma, S.; Sharma, S. Understanding Activation Functions in Neural Networks. *Int. J. Eng. Appl. Sci. Technol.* **2017**, *4*, 310–316.
56. Apicella, A.; Donnarumma, F.; Isgrò, F.; Prevete, R. A survey on modern trainable activation functions. *Neural Netw.* **2021**, *138*, 14–32. [CrossRef]
57. Al-Rikabi, H.M.H.; Al-Ja'afari, M.A.M.; Ali, A.H.; Abdulwahed, S.H. Generic model implementation of deep neural network activation functions using GWO-optimized SCPWL model on FPGA. *Microprocess. Microsyst.* **2020**, *77*, 103141. [CrossRef]

58. Boob, D.; Dey, S.S.; Lan, G. Complexity of training ReLU neural network. *Discret. Optim.* **2020**, 100620. [CrossRef]
59. Liu, B. Understanding the loss landscape of one-hidden-layer ReLU networks. *Knowl. Based Syst.* **2021**, *220*, 106923. [CrossRef]
60. Bouwmans, T.; Javed, S.; Sultana, M.; Jung, S.K. Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. *Neural Netw.* **2019**, *117*, 8–66. [CrossRef]

symmetry

MDPI

Article

Harris Hawks Optimization with Multi-Strategy Search and Application

Shangbin Jiao [1,2], Chen Wang [1,*], Rui Gao [1,2,3], Yuxing Li [1] and Qing Zhang [1]

[1] School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China; jiaoshangbin@xaut.edu.cn (S.J.); gaorui@bjwlxy.edu.cn (R.G.); liyuxing@xaut.edu.cn (Y.L.); zhangqing@xaut.edu.cn (Q.Z.)

[2] Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an University of Technology, Xi'an 710048, China

[3] School of Electronic & Electrical Engineering, Baoji University of Arts and Sciences, Baoji 721016, China

* Correspondence: 1180311029@stu.xaut.edu.cn

Abstract: The probability of the basic HHO algorithm in choosing different search methods is symmetric: about 0.5 in the interval from 0 to 1. The optimal solution from the previous iteration of the algorithm affects the current solution, the search for prey in a linear way led to a single search result, and the overall number of updates of the optimal position was low. These factors limit Harris Hawks optimization algorithm. For example, an ease of falling into a local optimum and the efficiency of convergence is low. Inspired by the prey hunting behavior of Harris's hawk, a multi-strategy search Harris Hawks optimization algorithm is proposed, and the least squares support vector machine (LSSVM) optimized by the proposed algorithm was used to model the reactive power output of the synchronous condenser. Firstly, we select the best Gauss chaotic mapping method from seven commonly used chaotic mapping population initialization methods to improve the accuracy. Secondly, the optimal neighborhood perturbation mechanism is introduced to avoid premature maturity of the algorithm. Simultaneously, the adaptive weight and variable spiral search strategy are designed to simulate the prey hunting behavior of Harris hawk to improve the convergence speed of the improved algorithm and enhance the global search ability of the improved algorithm. A numerical experiment is tested with the classical 23 test functions and the CEC2017 test function set. The results show that the proposed algorithm outperforms the Harris Hawks optimization algorithm and other intelligent optimization algorithms in terms of convergence speed, solution accuracy and robustness, and the model of synchronous condenser reactive power output established by the improved algorithm optimized LSSVM has good accuracy and generalization ability.

Keywords: Harris Hawks optimization algorithm; chaotic mapping; multi-strategy strategy; least squares support vector machine; synchronous condenser

Citation: Jiao, S.; Wang, C.; Gao, R.; Li, Y.; Zhang, Q. Harris Hawks Optimization with Multi-Strategy Search and Application. *Symmetry* **2021**, *13*, 2364. https://doi.org/10.3390/sym13122364

Academic Editors: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Received: 5 November 2021
Accepted: 30 November 2021
Published: 8 December 2021

1. Introduction

Along with the significant increase in the processing power of computer hardware and software, a large number of excellent meta-heuristics were created in the intelligent computing field [1–5]. Meta-heuristics are a large class of algorithms developed in contrast to optimization and heuristics. Optimization algorithms are dedicated to finding the optimal solution to a problem, but they are often difficult to implement due to the unresolvability of the problem [6,7]. Heuristic algorithms are dedicated to customizing algorithms through intuitive experience and problem information, but are often difficult to generalize due to their specialized nature. Compared to these two algorithms, meta-heuristic algorithms are more general and do not require deep adaptation to the problem, and although they do not guarantee optimal solutions, they can generally obtain optimal solutions under acceptable spatial and temporal conditions, although the degree of deviation from the optimal solution is difficult to estimate [8–12].

The main optimization strategies of meta-heuristic algorithms are summarized as follows: (1) diversification of exploration in a wide range to ensure the global optimal solution; and (2) intensification and exploitation in a local range to obtain an optimal solution as close to the optimal solution as possible [13]. The main difference between various meta-heuristic algorithms is how to strike a balance between the two. Almost all meta-heuristic algorithms have the following characteristics: (1) they are inspired by some phenomena in nature, such as simulated physics, biology, and biological behavior; (2) they use stochastic strategies; (3) they do not use the gradient resolution information of the objective function; and (4) they have several parameters that need to be adapted to the problem, and (5) they have good parallel and autonomous exploration. Meta-heuristic algorithms have been widely used in all aspects of social production and life. Many related research papers are published every year in the fields of production scheduling [14,15], engineering computing [16,17], management decision-making [18,19], machine learning (ML) [20,21], system control [22], and many other disciplines.

P-meta-heuristics are categorized into four main groups [23,24]: (1) simulated physical process algorithm, (2) evolutionary algorithm, (3) simulated swarm intelligence algorithm, and (4) human behavior [25–27]. Algorithms for simulating physical processes include simulated annealing (SA) [28], gravitational search algorithm [29] which simulates earth gravity, artificial chemical reaction optimization algorithm [30], heat transfer search, which simulates the heat transfer search process in thermodynamics [31], Gases Brownian motion optimization, which simulates the phenomenon of Brownian motion in physics [32], Henry gas solubility optimization, which simulates Henry gas solubility process [33]; and evolutionary algorithm. In 1975, American professor Holland proposed the genetic algorithm (GA) based on the Darwinian evolutionary theory and the mechanism of superiority and inferiority in nature. GA [34], evolution strategies [35] and differential evolution [36], genetic programming [37], and Biogeography-Based Optimizer [38]; simulated population intelligence algorithms: the Artificial Bee Colony (ABC) algorithm [39] based on the honey bee harvesting mechanisms, Firefly Algorithm based on the flickering behavior of fireflies [40], Beetle Antennae Search algorithm based on the foraging principle of aspen bark beetles [41], Grey Wolf Optimization (GWO) algorithm inspired by the hierarchy and predatory behavior of gray wolf packs [42], and Virus Colony Search algorithm [43], which is based on the proliferation and infection strategies of viruses to survive and reproduce in the cellular environment through host cells. Simulation of human behavior: Tabu Search [44], Socio Evolution and Learning Optimization [45], Teaching Learning Based Optimization [46], and Imperialist Competitive Algorithm [47].

Harris Hawks Optimization (HHO) [24] is a swarm intelligence optimization algorithm proposed by Heidari et al. in 2019 to simulate the prey hunting process of Harris's hawks in nature. The algorithm was inspired by the three phases of Harris's hawks' predatory behavior: search, search-exploitation conversion, and exploitation. The algorithm has a simple principle, fewer parameters, and better global search capability. Therefore, HHO has been applied in image segmentation [48], neural network training [49], motor control [50] and other fields. However, similar to other swarm intelligence optimization algorithms, HHO has the disadvantages of slow convergence speed, low optimization accuracy, and easily falls into local optimum when solving complex optimization problems. For example, the literature [51] used the information exchange mechanism to enhance the population diversity, thus improving the convergence speed of the HHO algorithm with information exchange (IEHHO) algorithm. The limitation of the IEHHO algorithm is how to set the parameters of the proposed algorithm. Zhang et al. [52] introduced an exponentially decreasing strategy to update the energy factor to increase the exploration and exploitation capability obtained by the relatively higher values of escaping energy; Elgamal et al. [53] made two improvements: (1) they applied chaotic mapping in the initialization phase of HHO; and (2) they used the SA algorithm as the current best solution to improve HHO exploitation; Shiming Song et al. [54] applied Gaussian mutation and a dimension decision strategy of the cuckoo search method into this algorithm to increase the HHO's performance. The mechanism of cuckoo search was useful in improving the convergence speed of the search agents as well as sufficient excavation of the solutions in the search area, while the Gaussian mutation strategy performed well in increasing the accuracy and jumping of the local optimum.

However, according to the no free lunch theory [55], one meta-heuristics algorithm cannot always perform as the best on all operations. The original HHO method could not fully balance the exploration and exploitation phases, which resulted in insufficient global search capability and slow convergence of the HHO method. To alleviate these adverse effects, we propose an improved algorithm model called chaotic multi-strategy search HHO (CSHHO), which introduces chaotic mapping and global search strategy, to solve single-objective optimization problems efficiently.

Here, the initialization phase of HHO is replaced by chaotic mapping, which allows the population initialization phase to be evenly distributed in the upper and lower bounds to enhance the population diversity, simultaneously enabling the population to approach the prey location faster, which accelerates the convergence speed of the algorithm. The adaptive weights are added to the position update formula in the exploration phase of HHO to dynamically adjust the influence of the global optimal solution. In the update phase of the HHO, the optimal neighborhood perturbation strategy is introduced to prevent the algorithm from falling into the local optimal solution and to solve the premature aging phenomenon.

To verify the superior performance of the CSHHO algorithm, this experiment first tests the effect of common chaotic mappings of the HHO algorithm's performance. The selected chaotic mappings are Sinusoidal, Tent, Kent, Cubic, Logistic, Gauss, and Circle, and the experimental results will show that Gauss chaotic mapping improves the accuracy of the HHO algorithm to the greatest extent. Second, the HHO algorithm based on Gauss chaotic mapping with multi-strategy search is tested. Then, it is compared with other classic and state-of-the-art algorithms on 23 classic test functions and 30 IEEE CEC2017 competition functions to verify the significant superiority of the proposed paradigm over other algorithms by Friedman test and Bonferroni–Holm corrected Wilcoxon signed-rank test. Finally, CSHHO is applied to model the reactive power output problem of a synchronous condenser based on LSSVM. The complete results will show that the effectiveness of the proposed optimizer is better than other models in the experiment.

The remainder of this paper is organized as follows: Section 2 introduces the basic theory and structure of the original HHO algorithm. Section 3 introduces the chaotic operator and Global search strategy to integrate it into the original optimizer. Section 4 conducts a full range of experiments on the proposed method and demonstrates the experimental results. It further discusses the proposed method based on experimental results. Section 5 applies the proposed method to the LSSVM-based synchronous condenser reactive power output problem. Finally, Section 6 summarizes the study and proposes research ideas for the future.

2. Harris Hawks Optimization Algorithm

The HHO algorithm is a swarm intelligence optimization algorithm that is widely used in solving optimization problems. The main idea of the algorithm is derived from the cooperative behavior and chasing strategy of Harris's hawk when catching prey in nature [24]. In the process of prey capture, the HHO algorithm is divided into two segments according to the physical energy E of the prey at the time of escape: exploration and exploitation phases, as shown in Figure 1. During the exploration phase, Harris's hawks randomly select a perching location to observe and monitor their prey.

$$X_i'(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X_i(t)|, q \geqslant 0.5 \\ X_{rabbit}(t) - X_m(t) - r_3[\mathbf{lb} + r_4(\mathbf{ub} - \mathbf{lb})], q < 0.5 \end{cases} \tag{1}$$

where $X_{rabbit}(t)$ and $X_{rand}(t)$ denote the position of the prey and the individual position at time t, respectively, and q is a random number between (0, 1), the average individual position:

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{2}$$

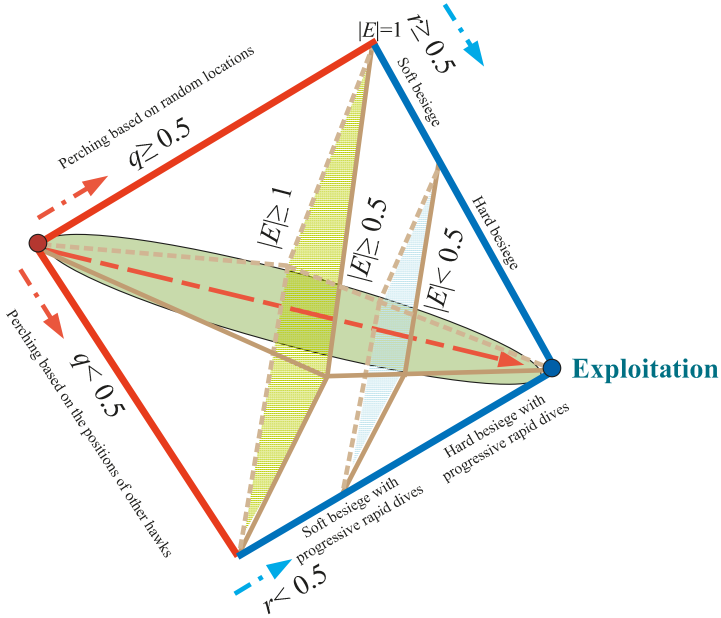

|E|=1
r≥ 0.5
Soft besiege
Perching based on random locations
q≥ 0.5
Hard besiege
|E|≥ 1
|E|≥ 0.5
|E|< 0.5
Exploitation
Perching based on the positions of other hawks
q< 0.5
Hard besiege with progressive rapid dives
Soft besiege with progressive rapid dives
r< 0.5

Figure 1. Description of each stage of HHO.

As the physical capacity of the prey decreases, the exploration phase changes to the exploitation phase, the prey's physical energy factor E is as follows:

$$E = 2E_0\left(1 - \frac{t}{T}\right) \tag{3}$$

In the exploitation phase, the Harris's hawk launches a surprise attack on the target prey found in the exploration phase, and the prey tries to escape when it encounters danger. Let the randomly generated prey escape probability be r, when $r < 0.5$ the prey successfully escapes; when $r > 0.5$ the prey does not successfully escape. According to the magnitude of r and $|E|$, four different location update strategies were proposed in the exploitation phase (see Table 1).

According to the position update condition in the HHO algorithm, the position of the Harris hawk is updated continuously, the fitness value was calculated according to the position of the Harris hawk, and if the fitness threshold was reached, the algorithm was finished. Otherwise, the algorithm continued to execute, and if the maximum number of iterations was reached, the algorithm was finished, and the optimal solution was obtained (See Figure 2).

Table 1. Exploitation phase of HHO algorithm.

Strategy	The value of r and $\lvert E \rvert$	
Soft besiege	$r \geq 0.5$ and $\lvert E \rvert \geq 0.5$	
	$X_i'(t+1) = \Delta X_i(t) - E(t)\lvert J X_{rabbit}\ (t) - X_i(t)\rvert$	(4)
	$\Delta X_i(t) = X_{rabbit}\ (t) - X_i(t)$	(5)
	$J = 2(1 - r_5)$	(6)
Hard besiege	$r \geq 0.5$ and $\lvert \mathrm{E} \rvert < 0.5$	
	$X_i'(t+1) = X_{rabbit}\ (t) - E(t)\lvert \Delta X_i(t)\rvert$	(7)
Soft besiege with progressive rapid dives	$r < 0.5$ and $\lvert E \rvert \geq 0.5$	
	$X_i'(t+1) = \begin{cases} Y & f(Y) < f(X_i(t)) \\ Z & f(Z) < f(X_i(t)) \end{cases}$	(8)
	$Y = X_{rabbit}\ (t) - E(t)\lvert J X_{rabbit}\ (t) - X_i(t)\rvert$	(9)
	$Z = Y + S \times LF(D)$	(10)
	$LF(\cdot) = 0.01 \times \frac{u}{\lvert v \rvert^{\frac{1}{\beta}}} \times \left(\frac{\Gamma(1+\beta) \times \sin(0.5\beta\pi)}{\Gamma(0.5(1+\beta)) \times \beta \times 2^{(\beta-1)/2}} \right)^{\frac{1}{\beta}}$	(11)
Hard besiege with progressive rapid dives	$r < 0.5$ and $\lvert E \rvert < 0.5$	
	$X_i'(t+1) = \begin{cases} Y & f(Y) < f(X_i(t)) \\ Z & f(Z) < f(X_i(t)) \end{cases}$	(12)
	$Y = X_{rabbit}\ (t) - E(t)\lvert J X_{rabbit}\ (t) - X_m(t)\rvert$	(13)

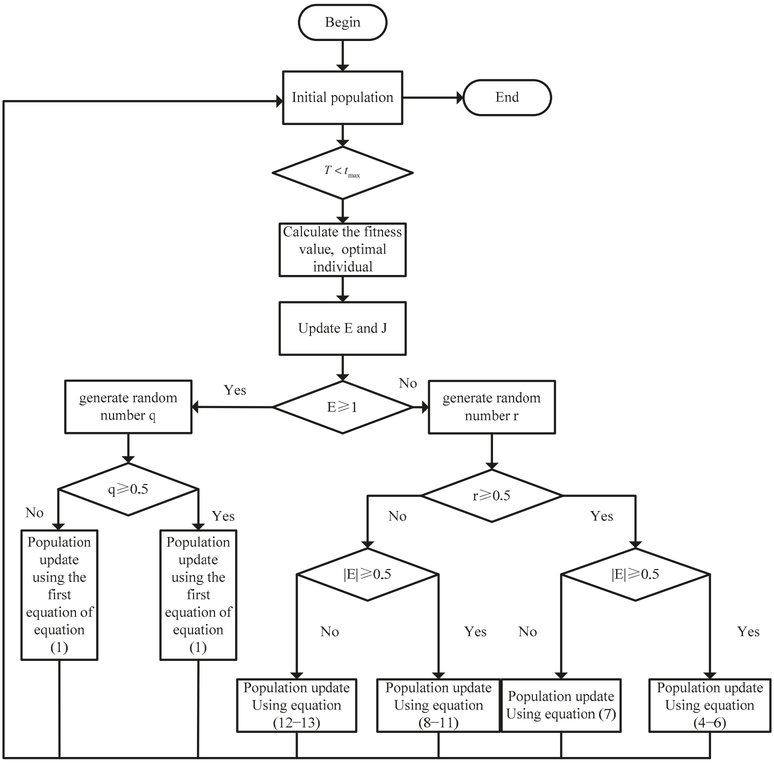

Begin
Initial population
End
$T < t_{\max}$
Calculate the fitness value, optimal individual
Update E and J
E⩾1
Yes
No
generate random number q
generate random number r
q⩾0.5
No
Yes
Population update using the first equation of equation (1)
Population update using the first equation of equation (1)
r⩾0.5
No
Yes
|E|⩾0.5
|E|⩾0.5
No
Yes
No
Yes
Population update Using equation (12–13)
Population update Using equation (8–11)
Population update Using equation (7)
Population update Using equation (4–6)

Figure 2. Flow chart of HHO algorithm.

3. HHO Algorithm Based on Multi-Search Strategy

3.1. Reasons for Improving the Basic HHO Algorithm

Harris's hawks generally gather high in trees to hunt for prey. In the process of hunting for prey, they often hover in a spiral to capture prey; when approaching prey, they rush towards their prey at a faster speed until the distance from the prey is small. They slow down and adjust their body posture to increase the probability of their catching prey [56–58]. This mechanism is important in the HHO algorithm. The exploration phase of the basic HHO algorithm uses Equations (1)–(3), where the optimal solution from the previous iteration of the algorithm affected the current solution and caused the algorithm to fall into a local optimum. The search for prey in a linear way led to a single search result. From all iterations of the algorithm, the optimal position of the current algorithm was only updated when the algorithm searched for a better solution than the current one, and the overall number of updates of the optimal position was low, which led to a decrease in the efficiency of the algorithm's search. In reality, when a Harris's hawk chases its prey, it hovers and descends in a spiral manner to catch its prey adaptively, showing better agility when hunting.

Here, the optimal neighborhood disturbance strategy was introduced to enhance the convergence speed of the algorithm and avoid premature maturity of the algorithm. The adaptive weighting and variable spiral position update strategies were introduced to enhance the global search capability of the algorithm by simulating the predation process of Harris's hawks in nature. To make the initial solution generated in the population initialization phase of the HHO algorithm cover the solution space as much as possible, we selected the best chaotic mapping method for HHO among seven commonly used chaotic mapping population initialization methods. It was used as the population initialization method to improve the algorithm. Hence, the above four methods are used to improve the global search capability of the HHO algorithm and to increase the speed of Harris Hawk's search for the optimal solution.

3.2. Chaotic Mapping

Chaotic is a deterministic stochastic method found in non-periodic, non-convergence and bounded nonlinear dynamic systems. Mathematically, chaotic is the randomness of a simple deterministic dynamic system, and a chaotic system is considered as the source of randomness. The essence of chaotic is obviously random and unpredictable, and it also has regularity [59].

As an important part of the population initialization algorithm, its result directly affects the convergence speed and quality of the algorithm [60,61]. For example, uniform distribution has more complete coverage of solution space than random distribution, and it is easier to obtain good initial solutions. A classical HHO algorithm uses random population initialization operation, which cannot cover the whole solution space. A chaotic sequence has ergodicity, randomness, and regularity in a certain range. Compared with random search, chaotic sequence searches the search space thoroughly with higher probability, which enables the algorithm to go beyond the local optimum and maintain the diversity of the population. Based on the above analysis, to obtain a good initial solution position and speed up the convergence of the population, seven common chaotic mappings Sinusoidal, Tent, Kent, Cubic, Logistic, Gauss, and Circle were selected [62–69] and used to initialize the population of HHO algorithm. The results were analyzed and the optimal one for the HHO algorithm selected as the population initialization method for the improved algorithm. The following were the mathematical formulas of the 10 chaotic mappings:

(1) Sinusoidal chaotic mapping:

$$x_{k+1} = P \cdot \sin(2\pi x_k) \tag{14}$$

where P was the control parameter, here $P = 2.3$, $x_0 = 0.7$, Equation (14) was written as

$$x_{k+1} = \sin(\pi x_k) \tag{15}$$

(2) Tent chaotic mapping

$$x_{k+1} = \begin{cases} 2x_k & x_k < 0.5 \\ 2(1-x_k) & x_k \geq 0.5 \end{cases} \tag{16}$$

(3) Kent chaotic mapping

$$\begin{cases} x_{k+1} = x_k/\mu & 0 < x_k < \mu \\ x_{k+1} = (1-x_k)/(1-\mu) & \mu \leqslant x_k < 1 \end{cases} \tag{17}$$

The control parameter $\mu \in (0,1)$, when $\mu = 0.5$, the system was Short Period State, $\mu = 0.5$ was not taken here. When using the chaotic mapping, the initial value x_0 had to not be the same as the system parameters μ, otherwise the system evolved into a periodic system. Here, we took $\mu = 0.4$.

(4) Cubic chaotic mapping

The standard Cubic chaotic mapping function was expressed as

$$x_{k+1} = bx_k^3 - cx_k \tag{18}$$

where b and c were the influence factors of chaotic mapping. The range of Cubic chaotic mapping was different for different values to b and c. When $c = 3$, the sequence generated by Cubic mapping was chaotic. Also when b = 1, $x_n \in (-2,2)$; when b = 4, $x_n \in (-1,1)$. Here, we took b = 4 and c = 3.

(5) Logistic chaotic mapping

$$x_{k+1} = Px_k(1-x_k) \tag{19}$$

when P = 4, the generation number of Logistic chaotic mapping was between (0, 1).

(6) Gauss chaotic mapping

$$x_{k+1} = \begin{cases} 0 & x_k = 0 \\ \frac{1}{x_k \bmod (1)} & \text{otherwise} \end{cases}$$
$$\frac{1}{x_k \bmod (1)} = \frac{1}{x_k} - \left|\frac{1}{x_k}\right| \tag{20}$$

(7) Circle chaotic mapping

$$x_{k+1} = \bmod\left(x_k + 0.2 - \left(\frac{0.5}{2\pi}\right)\sin(2\pi x_k), 1\right) \tag{21}$$

The three steps to initialize the population of the HHO using the seven chaotic mappings were:

Step 1: Randomly generate M Harris hawks in D-dimensional space, i.e., $Y = (y_1, y_2, y_3, \cdots \cdots, y_n)\ y_i \in (-1,1) i = 1,2,\cdots,n$.

Step 2: Iterate each dimension of each Harris hawk M times, resulting in M Harris hawks.

Step 3: After all Harris hawk iterations were completed, chaotic mapping (21) was applied to the solution space.

$$x_{id} = lb_d + (1 + y_{id}) \times \frac{ub_d - lb_d}{2} \tag{22}$$

where ub was the upper bound of the exploration space, lb the lower bound of the exploration space; the d-dimensional coordinates of the i-th Harris hawk were represented by y_{id}, which was generated using Equations (14)–(21); the coordinates of the i-th Harris hawk in the d-dimensional of the exploration space were x_{id}, which was generated using Equation (22).

Here, we first proposed the HHO algorithm based on seven different chaotic initialization strategies, respectively, chaotic initialization Harris hawks optimization (CIHHO) algorithm. Obviously, the implementation of CIHHO is basically the same as that of HHO, except that the initialization in Step 2 generates *m* individual Harris hawks using Equations (14)–(21), and then maps the positions of these *m* Harris hawks to the search space of the population using Equation (22).

3.3. Adaptive Weight

Inspired by the predation process of the Harris's hawk hunting strategy, we added an adaptive weight to the position update of Harris's hawk that changed with the number of iterations. In the early stage of the exploration phase of HHO, the influence of the optimal Harris's hawk position on the current individual position adjustment was weakened to improve the global search ability of the algorithm in the early stage. As the number of iterations increased, the influence of the optimal Harris's hawk position gradually increased, so that other Harris hawks could quickly converge to the optimal Harris hawk position and improve the convergence speed of the whole algorithm. According to the variation of the number of updates in the HHO algorithm, the adaptive weight composed of the number of iterations *t* were chosen as follows:

$$w(t) = 0.2\cos\left(\frac{\pi}{2} \cdot \left(1 - \frac{t}{t_{\max}}\right)\right) \tag{23}$$

Such adaptive weight $w(t)$ had a property of nonlinear variation between [0, 1], due to the variation property of the *cos* function between $[0, \frac{\pi}{2}]$, so that the weights were small at the beginning stage of the exploration phase, but changed slightly faster; at the end of the exploration phase their values were larger, but the speed of change would slow down, so that the convergence of the algorithm was fully guaranteed. The improved HHO algorithm position update formula is:

$$X(t+1) = \begin{cases} w(t)X_{rand}\,(t) - r_1|X_{rand}\,(t) - 2r_2X(t)| & q \geq 0.5 \\ w(t)(X_{rabbit}\,(t) - X_m(t)) - r_3(lb + r_4(ub - lb)) & q < 0.5 \end{cases} \tag{24}$$

The position update after the introduction of adaptive weights dynamically adjusted the weight size according to the increase of the number of iterations, so that the randomly selected Harris's hawk position $X_{rand}\,(t)$ and the optimal average Harris's hawk position $X_{rabbit}\,(t) - X_m(t)$ in the population guide the individual Harris's hawks differently at different times. As the number of iterations increased, the Harris's hawk population would move closer to the optimal position, and the larger weights would speed up the movement of Harris's hawk positions, which accelerated the convergence of the algorithm.

3.4. Variable Spiral Position Update

In the search phase of the HHO algorithm, Harris's hawk randomly searched for prey in two equal-opportunity strategies based on target location and its own location. However, in nature, Harris's hawks generally hover in a spiral shape to search for prey. To simulate the real process of prey search in nature, we introduced a variable spiral position update strategy in the search phase of the HHO algorithm, so that the Harris's hawk would adjust the distance of each position update according to the spiral shape between the target position and its own position (see Figure 3).

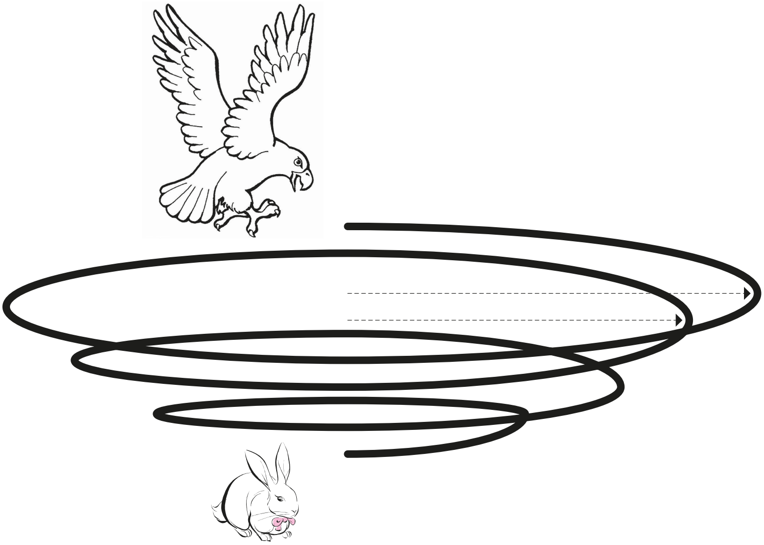


Figure 3. Spiral position update.

In the exploration phase of the HHO algorithm, Equation (1), a constant *b* was introduced to control the shape of the spiral; if this parameter was set to a constant, each time the Harris's hawk position updated a different spiral arc for speed adjustment would follow. However, if *b* was set to a constant value, the spiral movement of the Harris's hawk would be too singular when searching for prey, and it would follow a fixed spiral line to approach the target every time, which would easily fall into the misconception of local optimal solution and weaken the global exploration ability of the algorithm. To address this, we introduced the idea of variable spiral search to enable the Harris's hawk to develop more diverse search path strategies for location update and design the parameter *b* as a variable that changes with the number of iterations to dynamically adjust the shape of the spiral when the Harris's hawk explores, to increase the ability of the Harris's hawk to explore unknown areas; thus, improving the global search capability of the algorithm. After combining the adaptive weights, the new spiral position update was created (see Equation (25)).

The *b* parameter was designed based on the mathematical model of the spiral, and the spiral shape was dynamically adjusted by introducing the number of iterations on the basis of the original spiral model. The *b* parameter was designed in such a way that the spiral shape changed from large to small as the number of iterations increased. Early in the exploration phase of the HHO algorithm, the Harris's hawk searches the target with a larger spiral shape, the Harris hawk explores the global optimal solution as much as possible to improve the global optimal search capability of the algorithm; later in the exploration phase of the HHO algorithm, the Harris's hawk searched the target with a small spiral shape to improve the algorithm's search accuracy.

$$X(t+1) = \left\{ \begin{array}{c} w(t)X_{rand}\,(t) - \mathrm{b}|X_{rand}\,(t) - 2r_2X(t)| \quad q \geq 0.5 \\ w(t)(X_{rabbit}\,(t) - X_m(t)) - b(lb + r_4(ub - lb)) \quad q < 0.5 \\ b = e^{5(\pi(1-\frac{t}{t_{\max}}))} \end{array} \right. \tag{25}$$

3.5. Optimal Neighborhood Disturbance

When updating the position, the Harris's hawk generally takes the current optimal position as the target of this iteration. In the whole iteration, the optimal position is updated only when there is a better position; thus, the total number of updates was not many, which led to the inefficiency of the algorithm search. In this regard, the optimal neighborhood disturbance strategy was introduced to search the neighborhood of the optimal position randomly to find a better global value, which could not only improve the convergence speed of the algorithm, but also avoided premature maturity of the algorithm. The optimal position generated a random disturbance to increase its search of the nearby space, and the neighborhood disturbance formula was:

$$\widetilde{X}(t) = \begin{cases} X^*(t) + 0.5 \cdot \mathrm{h} \cdot X^*(t),\ g < 0.5 \\ X^*(t),\ g \geqslant 0.5 \end{cases} \tag{26}$$

where h and g were random numbers uniformly generated between [0, 1]; $X(t)$ was the new position generated. For the generated neighborhood positions, a greedy strategy was used to determine whether to keep them, and the formula was:

$$X^*(t) = \begin{cases} \widetilde{X}(t), f(\widetilde{X}(t)) < f(X^*(t)) \\ X^*(t), f(X^*(t)) \leqslant f(\widetilde{X}(t)) \end{cases} \tag{27}$$

where $f(x)$ was the position adaptation value of x. If the generated position was better than the original position, it would be replaced with the original position to make it the global optimum. Otherwise, the optimal position remained unchanged.

3.6. Computational Complexity

The computational complexity of the population initialization process of the classical HHO algorithm is O(N), and the computational complexity of the updated mechanism was O(T × N) + O(T × N × D), so the computational complexity of the classical HHO algorithm was O(N × (T + TD + 1)), where *T* was the maximum number of iterations and D the dimension of the specific problem. The computational complexity of the population initialization process of the CSHHO algorithm was O(ND), and the computational complexity of the update mechanism was the same as that of the classical HHO algorithm, so the computational complexity of the CSHHO algorithm was O(N × (T + TD + D)), where *T* was the maximum number of iterations, and D the dimensionality of the specific problem.

3.7. Algorithm Procedure

Algorithm 1 shows the procedure of the CSHHO optimization algorithm: See Algorithm 1.

Algorithm 1: CSHHO algorithm

```
Input: The population size N, maximum number of iterations T.
Output: The location of rabbit and its fitness value.
Using Seven chaotic maps to initialize the population:
Through chaotic variables $y_i^k \in [0, 1]$, $k = 1, 2, \ldots, M$. M indicates the initial population
dimension and Equations (14)–(21)generate initial chaotic vector;
Inverse mapping to get the initial population of the corresponding solution space through
Equation (22);
While stopping condition is not meet do
Calculate the fitness values of hawks;
      Set $X_{rabbit}$ as best location of rabbit;
      For each $X_i$ do
         Update the E using Equation (3);
         if $|E| \gg 1$ then
               Update the vector $X_i$ using Equations (25) and (2); Random generate parameters: $r_1$, $r_2$;
               $b = e^{5\times(\pi(1-\frac{t}{t_{\max}}))}$;
If $q \geq 0.5$ then
$X_i(t+1) = \omega(t) \times X_{rand}(t) - b \times |X_{rand}(t) - 2 \times r_2 \times X_i(t)|$;
end
if q < 0.5 then
$X_i(t+1) = \omega(t) \times (X_{rabbit}(t) - X_m(t)) - b \times |lb + r_4 \times (ub - lb)|$;
end
end
         if |E| < 1 then
               if $r \geq 0.5$ and |E| $\geq 0.5$ then
Update the vector $X_i$ using Equations (4)–(6);
end
               else if $r \geq 0.5$ and |E| < 0.5 then
                     Update the vector $X_i$ using Equation (7);
                end
               else if r < 0.5 and E |$\geq 0.5$
         Update the vector $X_i$ using Equations (8)–(11);
                end
               else if r < 0.5and |E| < 0.5
                     Update the vector $X_i$ using Equations (12) and (13);
                end
                end
end
Optimal neighborhood disturbance using Equations (27) and (28);
$\tilde{X}(t) = \begin{cases} X^*(t) + 0.5 \times h \times X^*(t), g < 0.5 \\ X^*(t), g > 0.5 \end{cases}$;
$X^*(t) = \begin{cases} \tilde{X}(t), f(\tilde{X}(t)) < f(X^*(t)) \\ X^*(t), f(X^*(t)) > f(\tilde{X}(t)) \end{cases}$;
end
Return $X_{rabbit}$;
```

4. Experiments and Discussion

In this section, to test and verify the performance of our optimizer proposed, namely CSHHO, a different category of experiments were designed. According to the randomness of the HHO algorithm, this section used a necessary and acceptable set of test functions to ensure that the superior results of the CSHHO algorithm did not happen by accident. Therefore, this section used two different benchmark test suites: classical 23 well-known benchmark functions [70,71] and standard IEEE CEC 2017 [72]. All experiments were as follows:

Experiment 1: First, seven chaotic mappings were used as the initialization method of HHO population and tested separately. Second, the seven data sets are analyzed, and the optimal chaotic mapping was selected as the population initialization method of the improved algorithm.

Experiment 2: First, on the basis of Experiment 1, a combination test of adaptive weighting mechanism, variable spiral position update and optimal neighborhood disturbance mechanism was executed. Second, we analyzed and compared the CSHHO algorithm with other recently proposed meta-heuristic algorithms such as HHO [24], WOA [38], SCA [73], and Chicken Swarm Optimization (CSO) [74]. Third, we analyzed and compared the CSHHO algorithm with developed advanced variants such as HHO with dimension decision logic and Gaussian mutation (GCHHO) [54] and Hybrid PSO Algorithm with Adaptive Step Search (DEPSOASS) [75] and Gravitational search algorithm with linearly decreasing gravitational constant (Improved GSA) [76] and Dynamic Generalized Opposition-based Learning Fruit Fly Algorithm (DGOBLFOA) [77]. Fourth, based on the third step, the IEEE CEC 2017 was used to perform an algorithm-based accuracy scalability test with test dimensions D = 50, D = 100.

To ensure the fairness of the experiments, the experiments were evaluated using the same parameters, and all population sizes N were set to 30, and dimension D was set to 30; each algorithm on each test instance was performed over 50 independent runs. In each run, the function error value $\log(F(x) - F(x^*))$, where *F(x)* was mean value found at all of the run, and $F(x^*)$ was the optimal value recorded in 23 benchmark functions. The average error (Mean) and standard deviation (Std) of the function error values were considered as two performance metrics for evaluating the performance of the algorithm in all runs. The experimental environment: CPU Intel(R) Xeon(R) CPU E5-2680 v3 (2.50 GHz), RAM 16.00 GB, MATLAB R2019b.

4.1. Benchmark Functions Verification

All experiments were performed using the classical 23 test functions [70,71] to test the performance of each algorithm in terms of convergence speed and search accuracy. These Benchmark functions were divided into three categories, including unimodal (UM) and multi-modal (MM). F1–F7 were the UM functions, which had unique global optimality and were used to test the exploitation performance of optimization algorithms. F8–F23 were the MM functions, which were used to test the exploration performance of the optimization algorithm and LO avoidance potentials. As the complexity of the test functions increased, the tested algorithms were more likely to fall into local optima, and all the test functions were used to evaluate the performance of the tested algorithms in various aspects. The convergence curves and test values of the corresponding test functions are given. Appendix A shows the classical 23 test functions.

IEEE CEC 2017 functions were also used in Experiment 2 to evaluate the scalability of CSHHO, other meta-heuristic algorithms, and developed HHO advanced variants. IEEE CEC 2017 Benchmark functions were classified into four categories, consisting of three UM functions (F1–F3), seven MM functions (F4–F10), 10 hybrid functions (F11–F20), and 10 composite functions (F21–F30). To evaluate the scalability of each algorithm more comprehensively, the dimensions of Benchmark functions were set to D = 50, D = 100, and Table 2 records the corresponding accuracy values. It also shows the function formulas for IEEE CEC 2017.

In addition, to compare the performance of various algorithms, the rank was used to rank the mean values of all the algorithms in the simulation experiment in the order of lowest to highest. The lower the rank, the better the algorithm was compared to other algorithms; conversely, the higher the rank, the worse the algorithm was compared to other algorithms. Wilcoxon signed-rank test [78] was used to detect whether there was a significant performance difference among all algorithms, the *p*-value was corrected by Bonferroni–Holm correction [79]; moreover, the Friedman test [80] was used to rank the superiority of all the algorithms.

Table 2. The information of IEEE CEC2017.

ID	Description	Type	Dimension	Range	Optimum
F1	Shifted and Rotated Bent Cigar Function	Unimodal	30, 50, 100	[−100, 100]	100
F2	Shifted and Rotated Zakharov function	Unimodal	30, 50, 100	[−100, 100]	300
F3	Shifted and Rotated Rosenbrock's function	Multimodal	30, 50, 100	[−100, 100]	400
F4	Shifted and Rotated Rastrigin's function	Multimodal	30, 50, 100	[−100, 100]	500
F5	Shifted and Rotated Expanded Scaffer's F6 function	Multimodal	30, 50, 100	[−100, 100]	600
F6	Shifted and Rotated Lunacek Bi-Rastrigin function	Multimodal	30, 50, 100	[−100, 100]	700
F7	Shifted and Rotated Non-Continuous Rastrigin's function	Multimodal	30, 50, 100	[−100, 100]	800
F8	Shifted and Rotated Lévy function	Multimodal	30, 50, 100	[−100, 100]	900
F9	Shifted and Rotated Schwefel's function	Multimodal	30, 50, 100	[−100, 100]	1000
F10	Hybrid Function 1 (N = 3)	Hybrid	30, 50, 100	[−100, 100]	1100
F11	Hybrid Function 2 (N = 3)	Hybrid	30, 50, 100	[−100, 100]	1200
F12	Hybrid Function 3 (N = 3)	Hybrid	30, 50, 100	[−100, 100]	1300
F13	Hybrid Function 4 (N = 4)	Hybrid	30, 50, 100	[−100, 100]	1400
F14	Hybrid Function 5 (N = 4)	Hybrid	30, 50, 100	[−100, 100]	1500
F15	Hybrid Function 6 (N = 4)	Hybrid	30, 50, 100	[−100, 100]	1600
F16	Hybrid Function 6 (N = 5)	Hybrid	30, 50, 100	[−100, 100]	1700
F17	Hybrid Function 6 (N = 5)	Hybrid	30, 50, 100	[−100, 100]	1800
F18	Hybrid Function 6 (N = 5)	Hybrid	30, 50, 100	[−100, 100]	1900
F19	Hybrid Function 6 (N = 6)	Hybrid	30, 50, 100	[−100, 100]	2000
F20	Composition Function 1 (N = 3)	Composition	30, 50, 100	[−100, 100]	2100
F21	Composition Function 2 (N = 3)	Composition	30, 50, 100	[−100, 100]	2200
F22	Composition Function 3 (N = 4)	Composition	30, 50, 100	[−100, 100]	2300
F23	Composition Function 4 (N = 4)	Composition	30, 50, 100	[−100, 100]	2400
F24	Composition Function 5 (N = 5)	Composition	30, 50, 100	[−100, 100]	2500
F25	Composition Function 6 (N = 5)	Composition	30, 50, 100	[−100, 100]	2600
F26	Composition Function 7 (N = 6)	Composition	30, 50, 100	[−100, 100]	2700
F27	Composition Function 7 (N = 6)	Composition	30, 50, 100	[−100, 100]	2800
F28	Composition Function 9 (N = 3)	Composition	30, 50, 100	[−100, 100]	2900
F29	Composition Function 10 (N = 3)	Composition	30, 50, 100	[−100, 100]	3000

We used the values of the Friedman test to rank all the algorithms involved in the comparison, if the values of the Friedman test were the same, then the rankings were averaged. Here, the Friedman test was performed on the classical 23 test functions, and the test values were recorded in the average ranking values (ARV) column.

4.2. Efficiency Analysis of the Improvement Strategy

First, in the population initialization phase, we selected the Gauss mapping, which had the highest impact on the accuracy of the HHO algorithm, as the population initialization method of CSHHO from seven commonly used chaotic mappings. Second, a global optimization strategy was used to optimize the HHO algorithm, which consisted of three components, including adaptive weight strategy, variable spiral update strategy and optimal neighborhood disturbance strategy. To verify the performance improvement of the HHO algorithm by the two improvements, six algorithms were used for comparison:

1. HHO without any modification, i.e., basic HHO;
2. WOA without any modification, i.e., basic WOA;
3. SCA without any modification, i.e., basic SCA;
4. CSO without any modification, i.e., basic CSO;
5. HHO with dimension decision logic and Gaussian mutation (GCHHO);
6. Hybrid PSO Algorithm with Adaptive Step Search (DEPSOASS);
7. Gravitational search algorithm with linearly decreasing gravitational constant (Improved GSA);
8. Dynamic Generalized Opposition-based Learning Fruit Fly Algorithm (DGOBLFOA).

4.2.1. Influence of Seven Common Chaotic Mappings on HHO Algorithm

In order to select the best effective chaotic mapping method among seven well-known chaotic mapping methods, which enables us to obtain the best initial solution position and speed up the convergence of the Harris Hawk algorithm population, sinusoidal chaotic mapping, Tent chaotic mapping, Kent chaotic mapping, Cubic chaotic mapping, Logistic chaotic mapping, Gauss chaotic mapping, Circle chaotic mapping were initialized to the population of the HHO algorithm, respectively, forming Sinusoidal-HHO, Tent-HHO, Kent-HHO, Cubic-HHO, Logistic-HHO, Gauss-HHO, Circle-HHO, and we compared the accuracy of these seven algorithms.

Table 3 presents the results of the seven algorithms for 23 classical test functions. The results included Best, Worst, Mean, Rank, and Std for each algorithm run 50 times independently.

Table 4 shows the Bonferroni–Holm corrected probability values p obtained from the Wilcoxon signed-rank test for the seven chaotic mapping HHO algorithms. Symbols "$+\backslash=\backslash-$" represent the number of algorithms that were better, similar, or worse than Gauss-HHO. The ARV at Table 5 is the value of the Friedman test for the seven chaotic mapping HHO algorithms.

Table 3 shows the data with better experimental results in bold. By analyzing the experimental results, we concluded that under the UM functions (F1–F7), Sinusoidal chaotic mapping achieved optimal results in F3, F5, F7 test functions, Circle chaotic mapping achieved optimal results in F1, F4 test functions, and Sinusoidal chaotic mapping had the most influence on the HHO algorithm, followed by Gauss chaotic mapping and Circle chaotic mapping. Under the MM functions (F8–F23), Gauss chaotic mapping had the most influence on the HHO algorithm. Circle chaotic mapping, Sinusoidal chaotic mapping, Tent chaotic mapping, and Kent chaotic mapping obtained the best results in F21, F15, F20, F13, and F23 test functions, respectively; the results of the seven chaotic mappings were compared in 23 test functions, The Gauss chaotic mapping obtained the most optimal solutions.

Table 4 shows the Bonferroni–Holm correction p-values of Wilcoxon signed rank test with 5% confidence level, "$+\backslash=\backslash-$"indicates whether Gauss-HHO was worse consistent or better with Circle-HHO, Sinusoidal-HHO, Tent-HHO, Kent-HHO, Cubic-HHO and Logistic-HHO. Analyzing the Bonferroni–Holm corrected p-value of Wilcoxon signed rank test and the value of "$+\backslash=\backslash-$" in each row of the table, better results were obtained using Gauss-HHO based among the 23 tested functions; the experimental results of HHO algorithms based on seven chaotic mappings, respectively, were evaluated comprehensively using Friedman's test at Table 5, compared with the other six chaotic mappings population initialization methods. Gauss-HHO obtained the best results in terms of average ranking, indicating that for the HHO optimization algorithm, Gauss chaotic mapping not only had the randomness, ergodicity and initial value sensitivity of the chaotic mapping itself, but also the population initialization of the HHO optimization algorithm using Gauss chaotic mapping. The Gauss chaos map was used to initialize the population of the HHO optimization algorithm, and to obtain a better optimization accuracy.

Table 3. Results of a comparison with seven chaotic mappings on HHO.

Benchmark		Circle	Sinusoidal	Tent	Kent	Cubic	Logistic	Gauss
F1	Mean	4.37×10^{-110}	2.73×10^{-96}	7.91×10^{-101}	3.68×10^{-99}	8.75×10^{-100}	7.28×10^{-99}	1.4×10^{-109}
	Std	3.05×10^{-109}	1.09×10^{-95}	3.91×10^{-100}	1.82×10^{-98}	4.06×10^{-99}	5.13×10^{-98}	9.69×10^{-109}
	Rank	1	7	3	5	4	6	2
	Best\Worst	2.15×10^{-108}\ 6.44×10^{-135}	4.36×10^{-95}\ 3.01×10^{-115}	2.21×10^{-99}\ 1.1×10^{-119}	9.39×10^{-98}\ 3.69×10^{-116}	2.11×10^{-98}\ 1.4×10^{-116}	3.63×10^{-97}\ 1.96×10^{-116}	6.85×10^{-108}\ 1.18×10^{-137}
F2	Mean	6.4×10^{-59}	3.88×10^{-53}	3.45×10^{-53}	7.39×10^{-53}	8.12×10^{-53}	8.59×10^{-53}	$\mathbf{8.05 \times 10^{-61}}$
	Std	3.12×10^{-58}	1.74×10^{-52}	1.5×10^{-52}	4.9×10^{-52}	5.03×10^{-52}	4.9×10^{-52}	$\mathbf{2.37 \times 10^{-60}}$
	Rank	2	4	3	5	6	7	**1**
	Best\Worst	1.97×10^{-57}\ 5.03×10^{-77}	1.20×10^{-51}\ 6.33×10^{-60}	8.88×10^{-52}\ 1.72×10^{-61}	3.47×10^{-51}\ 1.86×10^{-64}	3.56×10^{-51}\ 2.76×10^{-68}	3.4391×10^{-51}\ 3.91×10^{-62}	$\mathbf{1.0834 \times 10^{-59}}$\ $\mathbf{3.80 \times 10^{-68}}$
F3	Mean	2.90×10^{-86}	2.51×10^{-86}	5.22×10^{-82}	5.28×10^{-78}	3.08×10^{-83}	9.34×10^{-86}	7.29×10^{-78}
	Std	2.01×10^{-85}	9.38×10^{-86}	3.13×10^{-81}	3.73×10^{-77}	2.15×10^{-82}	3.43×10^{-85}	5.15×10^{-77}
	Rank	2	1	5	6	4	3	7
	Best\Worst	1.42×10^{-84}\ 8.08×10^{-104}	3.51×10^{-85}\ 9.55×10^{-104}	2.18×10^{-80}\ 1.01×10^{-108}	2.63×10^{-76}\ 5.02×10^{-101}	1.52×10^{-81}\ 2.2×10^{-107}	1.90×10^{-84}\ 5.74×10^{-105}	3.64×10^{-76}\ 1.58×10^{-107}
F4	Mean	8.48×10^{-56}	1.23×10^{-50}	7.91×10^{-51}	8.72×10^{-51}	3.65×10^{-50}	9.39×10^{-50}	9.3×10^{-53}
	Std	4.24×10^{-55}	4.28×10^{-50}	5.22×10^{-50}	3.6×10^{-50}	2.4×10^{-49}	6.51×10^{-49}	6.49×10^{-52}
	Rank	1	5	3	4	6	7	2
	Best\Worst	2.64×10^{-54}\ 9.93×10^{-69}	1.91×10^{-49}\ 1.3×10^{-55}	3.69×10^{-49}\ 2.77×10^{-59}	1.97×10^{-49}\ 4.06×10^{-58}	1.70×10^{-48}\ 2.16×10^{-60}	4.60×10^{-48}\ 4.39×10^{-58}	4.59×10^{-51}\ 1.13×10^{-65}
F5	Mean	8.55×10^{-2}	4.17×10^{-2}	5.42×10^{-2}	4.65×10^{-2}	7.29×10^{-2}	4.85×10^{-2}	8.33×10^{-2}
	Std	5.63×10^{-1}	4.90×10^{-3}	9.52×10^{-3}	7.56×10^{-3}	9.62×10^{-3}	8.98×10^{-3}	5.64×10^{-1}
	Rank	7	1	4	2	5	3	6
	Best\Worst	3.9896\ 4.3262×10^{-5}	0.021867\ 3.3302×10^{-6}	0.048302\ 8.4348×10^{-7}	0.0425\ 8.3416×10^{-7}	0.043385\ 3.4475×10^{-5}	0.051141\ 6.6967×10^{-6}	3.9896\ 6.6344×10^{-6}
F6	Mean	3.34×10^{-5}	7.87×10^{-5}	5.47×10^{-5}	4.72×10^{-5}	6.85×10^{-5}	4.45×10^{-5}	$\mathbf{2.48 \times 10^{-5}}$
	Std	6.12×10^{-5}	1.10×10^{-4}	9.45×10^{-5}	5.54×10^{-5}	$8.89 \times 10^{-5}6$	6.00×10^{-5}	$\mathbf{3.13 \times 10^{-5}}$
	Rank	2	7	5	4	6	3	**1**
	Best\Worst	3.42×10^{-4}\ 1.6931×10^{-8}	4.27×10^{-4}\ 7.8651×10^{-8}	3.71×10^{-4}\ 2.4303×10^{-9}	3.16×10^{-4}\ 3.9886×10^{-8}	4.33×10^{-4}\ 1.2627×10^{-8}	2.77×10^{-4}\ 2.6747×10^{-8}	$\mathbf{1.48 \times 10^{-4}}$\ $\mathbf{1.1473 \times 10^{-10}}$
F7	Mean	1.02×10^{-4}	7.60×10^{-5}	8.41×10^{-5}	9.33×10^{-5}	7.99×10^{-5}	7.62×10^{-5}	9.17×10^{-5}
	Std	9.77×10^{-5}	4.66×10^{-5}	1.02×10^{-4}	9.06×10^{-5}	8.17×10^{-5}	7.35×10^{-5}	8.83×10^{-5}
	Rank	7	1	4	6	3	2	5
	Best\Worst	4.09×10^{-4}\ 1.85×10^{-6}	1.92×10^{-4}\ 1.62×10^{-6}	5.44×10^{-4}\ 1.26×10^{-6}	3.88×10^{-4}\ 4.19×10^{-6}	5.07×10^{-4}\ 1.77×10^{-6}	4.39×10^{-4}\ 8.40×10^{-7}	4.36×10^{-4}\ 1.72×10^{-6}
F8	Mean	-1.26×10^{4}	-1.26×10^{4}	-1.26×10^{4}	-1.26×10^{4}	-1.25×10^{4}	-1.26×10^{4}	$\mathbf{-1.26 \times 10^{4}}$
	Std	52.1	0.232	0.453	0.508	0.378	0.226	**30.7**
	Rank	1	1	1	1	7	1	**1**
	Best\Worst	-1.23×10^{4}\ -1.25×10^{4}	-1.25×10^{4}\ -1.25×10^{4}	-1.25×10^{4}\ -1.25×10^{4}	-1.25×10^{4}\ -1.25×10^{4}	-9.90×10^{3}\ -1.25×10^{4}	-1.25×10^{4}\ -1.25×10^{4}	$\mathbf{-1.23 \times 10^{4}}$\ $\mathbf{-1.25 \times 10^{4}}$

Table 3. *Cont.*

Benchmark		Circle	Sinusoidal	Tent	Kent	Cubic	Logistic	Gauss
F9	Mean	0	0	0	0	0\0	0	**0**
	Std	0	0	0	0	0	0	**0**
	Rank	1	1	1	1	1	1	**1**
	Best\Worst	0\0	0\0	0\0	0\0	0\0	0\0	**0\0**
F10	Mean	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	$\mathbf{8.88 \times 10^{-16}}$
	Std	1.99×10^{-31}	1.99×10^{-31}	1.99×10^{-31}	1.99×10^{-31}	1.99×10^{-31}	1.99×10^{-31}	$\mathbf{1.99 \times 10^{-31}}$
	Rank	1	1	1	1	1	1	**1**
	Best\Worst	$8.88 \times 10^{-16} \backslash 8.88 \times 10^{-16}$	$8.88 \times 10^{-16} \backslash 8.88 \times 10^{-16}$	$8.88 \times 10^{-16} \backslash 8.88 \times 10^{-16}$	$8.88 \times 10^{-16} \backslash 8.88 \times 10^{-16}$	$8.88 \times 10^{-16} \backslash 8.88 \times 10^{-16}$	$8.88 \times 10^{-16} \backslash 8.88 \times 10^{-16}$	$\mathbf{8.88 \times 10^{-16} \backslash 8.88 \times 10^{-16}}$
F11	Mean	0	0	0	0	0	0	**0**
	Std	0	0	0	0	0	0	**0**
	Rank	1	1	1	1	1	1	**1**
	Best\Worst	0\0	0\0	0\0	0\0	0\0	0\0	**0\0**
F12	Mean	1.31×10^{-6}	2.79×10^{-6}	4.21×10^{-6}	2.75×10^{-6}	4.02×10^{-6}	4.18×10^{-6}	$\mathbf{1.18 \times 10^{-6}}$
	Std	1.49×10^{-6}	3.23×10^{-6}	8.73×10^{-6}	3.36×10^{-6}	5.88×10^{-6}	5.65×10^{-6}	$\mathbf{1.52 \times 10^{-6}}$
	Rank	2	4	7	3	5	6	**1**
	Best\Worst	$7.06 \times 10^{-6} \backslash 5.43 \times 10^{-9}$	$1.46 \times 10^{-5} \backslash 2.19 \times 10^{-8}$	$5.17 \times 10^{-5} \backslash 1.09 \times 10^{-8}$	$1.75 \times 10^{-5} \backslash 2.96 \times 10^{-9}$	$3.25 \times 10^{-5} \backslash 2.30 \times 10^{-9}$	$2.80 \times 10^{-5} \backslash 5.59 \times 10^{-9}$	$\mathbf{6.68 \times 10^{-6} \backslash 5.16 \times 10^{-10}}$
F13	Mean	2.66×10^{-4}	3.31×10^{-4}	3.91×10^{-5}	2.56×10^{-5}	4.15×10^{-5}	2.68×10^{-5}	4.62×10^{-4}
	Std	1.55×10^{-3}	2.07×10^{-3}	5.02×10^{-5}	4.26×10^{-5}	$5.82 \times 10^{-5}4$	3.68×10^{-5}	2.17×10^{-3}
	Rank	5	6	3	1	4	2	7
	Best\Worst	$0.010987 \backslash 1.4769 \times 10^{-7}$	$0.014637 \backslash 1.0906 \times 10^{-7}$	$0.00023743 \backslash 4.4639 \times 10^{-8}$	$0.00026746 \backslash 5.4412 \times 10^{-7}$	$0.00024897 \backslash 2.9199 \times 10^{-11}$	$0.00016986 \backslash 8.1121 \times 10^{-8}$	$0.010987 \backslash 5.0129 \times 10^{-8}$
F14	Mean	1.20	9.97×10^{-1}	9.98×10^{-1}	1.22	1.02	1.12	1.57
	Std	9.76×10^{-1}	5.61×10^{-16}	5.61×10^{-16}	9.82×10^{-1}	1.41×10^{-1}	7.10×10^{-1}	1.49
	Rank	5	1	2	6	3	4	7
	Best\Worst	$5.93 \backslash 9.98 \times 10^{-1}$	$9.98 \times 10^{-1} \backslash 9.98 \times 10^{-1}$	$9.98 \times 10^{-1} \backslash 9.98 \times 10^{-1}$	$5.93 \backslash 9.98 \times 10^{-1}$	$1.99 \backslash 9.98 \times 10^{-1}$	$5.93 \backslash 9.98 \times 10^{-1}$	$5.93 \backslash 9.98 \times 10^{-1}$
F15	Mean	3.24×10^{-4}	3.19×10^{-4}	3.22×10^{-4}	3.27×10^{-4}	3.72×10^{-4}	3.28×10^{-4}	3.25×10^{-4}
	Std	1.45×10^{-5}	1.03×10^{-5}	1.70×10^{-5}	1.93×10^{-5}	2.24×10^{-4}	1.79×10^{-5}	1.23×10^{-5}
	Rank	3	1	2	5	7	6	4
	Best\Worst	$3.82 \times 10^{-4} \backslash 3.08 \times 10^{-4}$	$3.48 \times 10^{-4} \backslash 3.08 \times 10^{-4}$	$3.72 \times 10^{-4} \backslash 3.08 \times 10^{-4}$	$3.92 \times 10^{-4} \backslash 3.08 \times 10^{-4}$	$1.63 \times 10^{-3} \backslash 3.08 \times 10^{-4}$	$3.86 \times 10^{-4} \backslash 3.08 \times 10^{-4}$	$3.95 \times 10^{-4} \backslash 3.08 \times 10^{-4}$
F16	Mean	−1.03	−1.03	−1.03	−1.03	−1.03	−1.03	**−1.03**
	Std	1.35×10^{-15}	1.35×10^{-15}	1.35×10^{-15}	1.35×10^{-15}	1.35×10^{-15}	1.35×10^{-15}	$\mathbf{1.35 \times 10^{-15}}$
	Rank	1	1	1	1	1	1	**1**
	Best\Worst	−1.03\−1.03	−1.03\−1.031	−1.03\−1.03	−1.03\−1.03	−1.03\−1.03	−1.03\−1.03	**−1.03\−1.03**

Table 3. *Cont.*

Benchmark		Circle	Sinusoidal	Tent	Kent	Cubic	Logistic	Gauss
F17	Mean	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	**3.9×10^{-9}**
	Std	7.97×10^{-8}	4.58×10^{-8}	2.57×10^{-7}	1.65×10^{-7}	3.38×10^{-7}	7.58×10^{-8}	**4.17×10^{-8}**
	Rank	2	2	2	2	2	2	**1**
	Best\Worst	$3.97 \times 10^{-1}\backslash$ 3.97×10^{-1}	$3.97 \times 10^{-1}\backslash$ 3.97×10^{-1}	$3.97 \times 10^{-1}\backslash$ 3.97×10^{-1}	$3.97 \times 10^{-1}\backslash$ 3.97×10^{-1}	$3.97 \times 10^{-1}\backslash$ $0.3.97 \times 10^{-1}$	$3.97 \times 10^{-1}\backslash$ $0.3.97 \times 10^{-1}$	**$3.97 \times 10^{-1}\backslash$ $0.3.97 \times 10^{-1}$**
F18	Mean	3	3	3	3	3	3	**3**
	Std	1.41×10^{-8}	1.31×10^{-7}	1.98×10^{-8}	5.48×10^{-8}	1.41×10^{-8}	7.00×10^{-8}	**4.63×10^{-8}**
	Rank	1	1	1	1	1	1	**1**
	Best\Worst	$3\backslash 3$	$3\backslash 3$	$3\backslash 3$	$3\backslash 3$	$3\backslash 3$	$3\backslash 3$	**$3\backslash 3$**
F19	Mean	-3.86	-3.86	-3.86	-3.86	-3.86	-3.86	**-3.86**
	Std	2.09×10^{-3}	2.52×10^{-3}	2.35×10^{-3}	2.43×10^{-3}	1.61×10^{-3}	3.02×10^{-3}	**1.32×10^{-3}**
	Rank	1	1	1	1	1	1	**1**
	Best\Worst	$-3.85\backslash-3.86$	$-3.85\backslash-3.86$	$-3.85\backslash-3.86$	$-3.85\backslash-3.86$	$-3.86\backslash-3.86$	$-3.85\backslash-3.86$	**$-3.86\backslash-3.86$**
F20	Mean	-3.12	-3.12	-3.14	-3.12	-3.13	-3.11	-3.12
	Std	1.10×10^{-1}	1.12×10^{-1}	1.10×10^{-1}	1.10×10^{-1}	1.10×10^{-1}	1.12×10^{-1}	9.95×10^{-2}
	Rank	3	3	1	3	2	7	3
	Best\Worst	$-2.88\backslash-3.31$	$-2.80\backslash-3.30$	$-2.89\backslash-3.30$	$-2.86\backslash-3.30$	$-2.90\backslash-3.30$	$-2.82\backslash-3.30$	$-2.81\backslash-3.27$
F21	Mean	-8.77	-7.05	-7.04	-8.75	-5.36	-7.03	-8.45
	Std	2.23	2.47	2.46	2.21	1.21	2.45	2.36
	Rank	1	4	5	2	7	6	3
	Best\Worst	$-5.05\backslash-1.01 \times 10^{1}$	$-5.05\backslash-1.01 \times 10^{1}$	$-5.05\backslash-1.01 \times 10^{1}$	$-5.05\backslash-1.01 \times 10^{1}$	$-5.04\backslash-1.01 \times 10^{1}$	$-5.05\backslash-1.01 \times 10^{1}$	$-5.05\backslash-1.01 \times 10^{1}$
F22	Mean	-8.44	-8.62	-7.06	-8.95	-5.49	-7.04	**-9.06**
	Std	2.54	2.45	2.55	2.32	1.38	2.53	**1.26**
	Rank	4	3	5	2	7	6	**1**
	Best\Worst	$-5.08\backslash-1.04 \times 10^{1}$	$-5.07\backslash-1.04 \times 10^{1}$	$-5.08\backslash-1.04 \times 10^{1}$	$-5.09\backslash-1.04 \times 10^{1}$	$-5.08\backslash-1.04 \times 10^{1}$	$-5.08\backslash-1.04 \times 10^{1}$	**$-5.09\backslash-1.04 \times 10^{1}$**
F23	Mean	-9.27	-8.61	-6.84	-9.54	-5.38	-7.34	-8.92
	Std	2.23	2.53	2.52	1.95	1.33	2.63	2.4
	Rank	2	4	6	1	7	5	3
	Best\Worst	$-5.12\backslash-1.05 \times 10^{1}$	$-5.123\backslash-1.05 \times 10^{1}$	$-5.12\backslash-1.05 \times 10^{1}$	$-5.12\backslash-1.05 \times 10^{1}$	$-2.41\backslash-1.05 \times 10^{1}$	$-5.13\backslash-1.05 \times 10^{1}$	$-5.12\backslash-1.05 \times 10^{1}$

Table 4. The Bonferroni–Holm corrected P-values of Wilcoxon's signed-rank test.

Benchmark	Gauss	Circle		Sinusoidal		Tent		Kent		Cubic		Logistic	
		Corrected p-Value	+\=\−	Corrected p-Value	+\=\−	Corrected p-Value	+\=\−	Corrected p-Value	+\=\−	Corrected p-Value	+\=\−	Corrected p-Value	+\=\−
F1	N/A	2.12×10^{1}	28\0\22	1.043×10^{-7}	50\0\0	5.40×10^{-7}	48\0\2	1.04×10^{-7}	50\0\0	1.08×10^{-7}	49\0\1	6.77×10^{-7}	48\0\2
F2	N/A	3.05×10^{1}	26\0\24	1.043×10^{-7}	50\0\0	1.20×10^{-7}	49\0\1	1.08×10^{-7}	49\0\1	1.20×10^{-7}	48\0\2	1.03×10^{-7}	50\0\0
F3	N/A	3.23×10^{1}	25\0\25	3.20×10^{1}	24\0\26	2.27×10^{1}	29\0\21	2.01×10^{0}	32\0\18	6.39×10^{0}	31\0\19	8.66×10^{0}	28\0\22
F4	N/A	1.64×10^{1}	23\0\27	1.02×10^{-5}	47\0\3	2.74×10^{-5}	46\0\4	3.24×10^{-7}	48\0\2	3.71×10^{-5}	46\0\4	3.61×10^{-7}	48\0\2
F5	N/A	2.02×10^{1}	30\0\20	2.79×10^{1}	27\0\23	2.79×10^{1}	27\0\23	2.85×10^{1}	26\0\24	6.12×10^{0}	31\0\19	2.59×10^{1}	28\0\22
F6	N/A	2.66×10^{1}	23\0\27	4.67×10^{-2}	36\0\14	8.66×10^{0}	33\0\17	1.74×10^{0}	32\0\18	2.25×10^{-2}	37\0\13	6.18×10^{0}	32\0\18
F7	N/A	3.34×10^{1}	26\0\24	3.13×10^{1}	28\0\22	3.22×10^{1}	23\0\27	2.72×10^{1}	23\0\27	3.26×10^{1}	23\0\27	3.12×10^{1}	24\0\26
F8	N/A	1.64×10^{1}	20\0\30	3.26×10^{1}	24\1\25	2.03×10^{1}	21\0\29	3.27×10^{1}	24\0\26	3.25×10^{1}	28\1\21	2.71×10^{1}	20\1\29
F9	N/A	2.49×10^{1}	0\50\0	2.10×10^{1}	0\50\0	1.70×10^{1}	0\50\0	1.30×10^{1}	0\50\0	9.00×10^{0}	0\50\0	5.00×10^{0}	0\50\0
F10	N/A	2.40×10^{1}	0\50\0	2.00×10^{1}	0\50\0	1.60×10^{1}	0\50\0	1.20×10^{1}	0\50\0	8.00×10^{0}	0\50\0	4.00×10^{0}	0\50\0
F11	N/A	2.30×10^{1}	0\50\0	1.90×10^{1}	0\50\0	1.50×10^{1}	0\50\0	1.10×10^{1}	0\50\0	7.00×10^{0}	0\50\0	3.00×10^{0}	0\50\0
F12	N/A	3.02×10^{1}	31\0\19	3.32×10^{-1}	33\0\17	5.13×10^{-1}	36\0\14	9.84×10^{-1}	31\0\19	1.26×10^{-2}	37\0\13	6.20×10^{-2}	36\0\14
F13	N/A	2.84×10^{1}	22\0\28	2.91×10^{1}	26\0\24	8.84×10^{0}	29\0\21	3.15×10^{1}	23\0\27	2.71×10^{1}	27\0\23	3.29×10^{1}	22\0\28
F14	N/A	1.26×10^{1}	2\39\9	4.30×10^{-1}	0\41\9	4.30×10^{-1}	0\41\9	1.32×10^{1}	3\39\8	1.23×10^{0}	1\40\9	4.88×10^{0}	1\40\9
F15	N/A	2.49×10^{1}	20\0\30	5.13×10^{-1}	14\0\36	1.31×10^{1}	18\0\32	2.79×10^{1}	21\0\29	1.99×10^{1}	29\0\21	2.27×10^{1}	27\0\23
F16	N/A	2.20×10^{1}	0\50\0	1.80×10^{1}	0\50\0	1.40×10^{1}	0\50\0	1.00×10^{1}	0\50\0	6.00×10^{0}	0\50\0	2.00×10^{0}	0\50\0
F17	N/A	3.23×10^{1}	10\28\12	2.68×10^{1}	11\28\11	2.20×10^{1}	12\27\11	9.10×10^{0}	20\21\9	9.18×10^{0}	22\17\11	3.25×10^{1}	8\31\11
F18	N/A	3.22×10^{1}	0\48\2	3.28×10^{1}	5\42\3	3.06×10^{1}	2\45\3	3.32×10^{1}	5\42\3	3.20×10^{1}	0\48\2	2.76×10^{1}	2\45\3
F19	N/A	3.28×10^{1}	25\0\25	1.32×10^{1}	30\0\20	3.23×10^{1}	27\0\23	3.26×10^{1}	25\0\25	2.92×10^{1}	25\0\25	3.24×10^{1}	22\0\28
F20	N/A	3.34×10^{1}	25\0\25	3.00×10^{1}	23\0\27	2.12×10^{1}	20\0\30	3.00×10^{1}	23\0\27	3.24×10^{1}	24\0\26	3.27×10^{1}	24\0\26
F21	N/A	2.03×10^{1}	21\0\29	2.15×10^{0}	32\0\18	2.87×10^{-1}	37\0\13	2.85×10^{1}	28\0\22	6.425×10^{-5}	40\0\10	8.45×10^{-2}	38\0\12
F22	N/A	2.20×10^{1}	29\0\21	1.89×10^{1}	30\0\20	4.39×10^{-2}	37\0\13	2.79×10^{1}	28\0\22	3.068×10^{-7}	47\0\3	1.22×10^{-2}	39\0\11
F23	N/A	2.75×10^{1}	23\0\27	3.05×10^{1}	28\0\22	2.31×10^{-1}	31\0\19	3.26×10^{1}	24\0\26	$9.225 \times 10-6$	40\0\10	1.24×10^{0}	30\0\20

Table 5. Average ranks obtained by each method in the Friedman test.

Chaotic Mapping	Average Rankings
Gauss	**3.50**
Circle	3.74
Sinusoidal	3.89
Tent	3.85
Kent	4.85
Cubic	4.57
Logistic	3.61

4.2.2. Comparison with Conventional Techniques

The Gauss mapping was used for population initialization. Then the adaptive weight mechanism, variable spiral position update mechanism, and adaptive neighborhood disturbance mechanism were introduced to form the CSHHO algorithm. In order to verify the effectiveness of the CSHHO algorithm against the emerging swarm intelligence optimization algorithms in recent years. In this subsection, the CSHHO algorithm is compared with recently published meta-heuristics, including HHO [24], WOA [24], SCA [65] and CSO [66] to calculate the average precision mean and stability Std of each algorithm. The performance of CSHHO was tested against other optimization algorithms using a nonparametric test: the Bonferroni–Holm corrected Wilcoxon signed rank test. Finally, the non-parametric test method (i.e., the Friedman test) was used to calculate the ARV values of all the participating algorithms and rank them together. As in Experiment 1, this experiment was also based on the test set of 23 classical test functions (see Table 1). The details of the experiment were consistent with the description at the beginning of this section, and Table 6 shows the detailed experimental results. Additionally, Table 7 gives the corrected Wilcoxon signed-rank test based on the 5% confidence level, the "$+\backslash=\backslash-$" value: the number of CSHHO results that were worse, similar, better or than the comparison algorithm for each test function run 50 times, the result based on the Friedman test was at table.

The optimal results of the tested algorithms under the current Benchmark function are marked in bold. Analyzing the data in Table 6, CSHHO had a strong optimization capability compared to the traditional optimization algorithms. Under the MM functions (F8–F23), CSHHO obtained good results with the best optimization results under the Benchmarks of F9–F13, F15–F19, F21–F23, and CSHHO explored the most optimal region of the above Benchmark and outperformed the other compared optimization algorithms in terms of search performance. The CSHHO explored the above Benchmark optimal regions and outperformed the other participating optimization algorithms in terms of exploration performance. It tied for first place in Benchmark F9, F11, and F11. This showed that CSHHO had strong exploration ability and LO avoidance potentials.

Table 7 was analyzed to determine if there was a significant difference between the other algorithms and CSHHO. The "$+\backslash=\backslash-$" column indicates the number of results that are less than, similar to, or greater than CSHHO for each of the HHO, WOA, SCA, and CSO algorithms run 50 times in each test function. CSHHO has 21 test functions with better results than HHO, CSHHO has 22 test functions with better results than WOA, CSHHO has 23 test functions with better results than SCA, CSHHO has 22 test functions with better results than CSO; the results of the corrected 5% confidence level Wilcoxon signed rank test were analyzed. If the p-value was greater than 0.05, the algorithm was considered to be the same as CSHHO; otherwise, it was considered to be significantly different. In most cases the p-values of Wilcoxon signed-rank test < 0.5, indicating that the CSHHO algorithm was significantly different from the other compared algorithms, all results were corrected by Bonferroni–Holm correction; At Table 8, analysis of the Friedman test value showed that the value of CSHHO was 2.57 lower than the traditional optimization algorithm. The CSHHO algorithm's performance was better than other meta-heuristic algorithms.

Table 6. Results of a comparison with classic meta-heuristic algorithms.

Benchmark		CSHHO	HHO	WOA	SCA	CSO
F1	Mean	$\mathbf{1.96 \times 10^{-113}}$	1.24×10^{-88}	6.15×10^{-74}	2.92×10^{-13}	2.82×10^{-19}
	Std	$\mathbf{1.38 \times 10^{-112}}$	8.75×10^{-88}	2.79×10^{-73}	1.33×10^{-12}	7.98×10^{-19}
	Rank	**1**	2	3	5	4
	Best\Worst	$\mathbf{1.44 \times 10^{-138} \backslash 9.78 \times 10^{-112}}$	$5.93 \times 10^{-111} \backslash 6.19 \times 10^{-87}$	$1.16 \times 10^{-87} \backslash 1.85 \times 10^{-72}$	$2.12 \times 10^{-20} \backslash 8.05 \times 10^{-12}$	$4.81 \times 10^{-26} \backslash 4.63 \times 10^{-18}$
F2	Mean	$\mathbf{1.66 \times 10^{-59}}$	1×10^{-49}	8.97×10^{-51}	1.04×10^{-10}	1.05×10^{-18}
	Std	$\mathbf{1.06 \times 10^{-58}}$	4.8×10^{-49}	4.65×10^{-50}	2.43×10^{-10}	2.76×10^{-18}
	Rank	**1**	3	2	5	4
	Best\Worst	$\mathbf{7.47 \times 10^{-58} \backslash 8.54 \times 10^{-70}}$	$1.86 \times 10^{-74} \backslash 1.28 \times 10^{-73}$	$4.54 \times 10^{4} \backslash 1.25 \times 10^{4}$	$2.19 \times 10^{-4} \backslash 1.31 \times 10^{-3}$	$6.13 \times 10^{3} \backslash 3.20 \times 10^{3}$
F3	Mean	$\mathbf{4.96 \times 10^{-88}}$	1.86×10^{-74}	4.54×10^{4}	0.000219	6.13×10^{3}
	Std	$\mathbf{2.81 \times 10^{-87}}$	1.28×10^{-73}	1.25×10^{4}	1.31×10^{-3}	3.2×10^{3}
	Rank	**1**	2	5	3	4
	Best\Worst	$\mathbf{5.85 \times 10^{-108} \backslash 1.96 \times 10^{-86}}$	$4.26 \times 10^{-99} \backslash 9.05 \times 10^{-73}$	$1.81 \times 10^{4} \backslash 7.29 \times 10^{4}$	$1.75 \times 10^{-12} \backslash 9.26 \times 10^{-3}$	$3.04 \times 10^{2} \backslash 1.65 \times 10^{4}$
F4	Mean	$\mathbf{1.05 \times 10^{-55}}$	7.88×10^{-49}	53.4	7.27×10^{-5}	21.6
	Std	$\mathbf{6.13 \times 10^{-55}}$	4.62×10^{-48}	2.82×10^{1}	1.38×10^{-4}	1.28×10^{1}
	Rank	**1**	2	5	3	4
	Best\Worst	$\mathbf{1.83 \times 10^{-68} \backslash 4.25 \times 10^{-54}}$	$2.8 \times 10^{-58} \backslash 3.25 \times 10^{-47}$	$9.79 \times 10^{-1} \backslash 8.84 \times 10^{1}$	$2.56 \times 10^{-7} \backslash 6.9 \times 10^{-4}$	$0.0748 \backslash 4.11 \times 10^{1}$
F5	Mean	$\mathbf{8.57 \times 10^{-3}}$	1.02×10^{-2}	2.79×10^{1}	1.11×10^{1}	4.77×10^{1}
	Std	$\mathbf{2.23 \times 10^{-2}}$	1.43×10^{-2}	4.98×10^{-1}	2.78×10^{1}	9.63×10^{1}
	Rank	**1**	2	4	3	5
	Best\Worst	$\mathbf{1.29 \times 10^{-8} \backslash 0.132}$	$1.25 \times 10^{-5} \backslash 0.0723$	$26.9 \backslash 28.8$	$6.48 \backslash 204$	$27.1 \backslash 583$
F6	Mean	$\mathbf{6.73 \times 10^{-7}}$	1.59×10^{-4}	0.374	0.377	3.73
	Std	$\mathbf{1.85 \times 10^{-6}}$	2.76×10^{-4}	2.44×10^{-1}	1.42×10^{-1}	4.16×10^{-1}
	Rank	**1**	2	3	4	5
	Best\Worst	$\mathbf{4.40 \times 10^{-14} \backslash 8.8 \times 10^{-6}}$	$4.85 \times 10^{-9} \backslash 0.00181$	$0.0422 \backslash 1.25$	$0.117 \backslash 0.627$	$3.05 \backslash 4.99$
F7	Mean	$\mathbf{5.98 \times 10^{-5}}$	2.02×10^{-4}	3.67×10^{-3}	1.84×10^{-3}	9.11×10^{-2}
	Std	$\mathbf{7.15 \times 10^{-5}}$	3.26×10^{-4}	3.79×10^{-3}	2.02×10^{-3}	1.82×10^{-1}
	Rank	**1**	2	4	3	5
	Best\Worst	$\mathbf{2.03 \times 10^{-7} \backslash 4.65 \times 10^{-4}}$	$3.13 \times 10^{-7} \backslash 2.10 \times 10^{-3}$	$9.46 \times 10^{-6} \backslash 1.75 \times 10^{-2}$	$6.53 \times 10^{-5} \backslash 1.02 \times 10^{-2}$	$2.66 \times 10^{-3} \backslash 1.07$
F8	Mean	-1.24×10^{4}	-1.26×10^{4}	-1.04×10^{4}	-2.24×10^{3}	-7.12×10^{3}
	Std	2.71×10^{2}	9.50×10^{1}	1.67×10^{3}	1.52×10^{2}	6.62×10^{2}
	Rank	2	1	3	5	4
	Best\Worst	$-1.26 \times 10^{4} \backslash -1.15 \times 10^{4}$	$-1.26 \times 10^{4} \backslash -1.19 \times 10^{4}$	$-1.26 \times 10^{4} \backslash -7.76 \times 10^{3}$	$-2.61 \times 10^{3} \backslash -1.81 \times 10^{3}$	$-8.89 \times 10^{3} \backslash -5.91 \times 10^{3}$
F9	Mean	**0**	0	3.41×10^{-15}	0.572	0.725
	Std	**0**	0	1.78×10^{-14}	2.03	3.34
	Rank	**1**	1	3	4	5
	Best\Worst	$\mathbf{0 \backslash 0}$	$0 \backslash 0$	$0 \backslash 1.14 \times 10^{-13}$	$0 \backslash 1.12 \times 10^{1}$	$0 \backslash 2.11 \times 10^{1}$
F10	Mean	$\mathbf{8.88 \times 10^{-16}}$	8.88×10^{-16}	4.3×10^{-15}	5.46×10^{-8}	9.39×10^{-11}
	Std	$\mathbf{1.99 \times 10^{-31}}$	1.99×10^{-31}	2.68×10^{-15}	1.41×10^{-7}	2.5×10^{-10}
	Rank	**1**	1	3	5	4
	Best\Worst	$\mathbf{8.88 \times 10^{-16} \backslash 8.88 \times 10^{-16}}$	$8.88 \times 10^{-16} \backslash 8.88 \times 10^{-16}$	$8.88 \times 10^{-16} \backslash 7.99 \times 10^{-15}$	$6.45 \times 10^{-11} \backslash 7.1 \times 10^{-7}$	$2.82 \times 10^{-13} \backslash 1.66 \times 10^{-9}$
F11	Mean	**0**	0	1.97×10^{-2}	9.43×10^{-2}	1.22×10^{-2}

Table 6. *Cont.*

Benchmark		CSHHO	HHO	WOA	SCA	CSO
	Std	**0**	0	7.55×10^{-2}	1.56×10^{-1}	4.49×10^{-2}
	Rank	**1**	1	4	5	3
	Best\Worst	$\mathbf{0\backslash 0}$	$0\backslash 0$	$0\backslash 4.34 \times 10^{-1}$	$0\backslash 7.65 \times 10^{-1}$	$0\backslash 2.47 \times 10^{-1}$
F12	Mean	$\mathbf{5.26 \times 10^{-7}}$	9.97×10^{-6}	2.69×10^{-2}	7.80×10^{-2}	229
	Std	$\mathbf{1.2 \times 10^{-6}}$	1.36×10^{-5}	0.04	2.80×10^{-2}	1.1×10^{3}
	Rank	**1**	2	3	4	5
	Best\Worst	$\mathbf{1.68 \times 10^{-11}\backslash 5 \times 10^{-6}}$	$3.07 \times 10^{-8}\backslash 7.06 \times 10^{-5}$	$4.58 \times 10^{-3}\backslash 2.64 \times 10^{-1}$	$3.41 \times 10^{-2}\backslash 1.63 \times 10^{-1}$	$1.57 \times 10^{-1}\backslash 7.34 \times 10^{3}$
F13	Mean	$\mathbf{1.64 \times 10^{-5}}$	1.26×10^{-4}	5.05×10^{-1}	2.64×10^{-1}	5.48
	Std	$\mathbf{2.86 \times 10^{-5}}$	2.49×10^{-4}	2.57×10^{-1}	8.66×10^{-2}	2.07×10^{1}
	Rank	**1**	2	4	3	5
	Best\Worst	$6.14 \times 10^{-10}\backslash 1.23 \times 10^{-4}$	$3.21 \times 10^{-7}\backslash 1.31 \times 10^{-3}$	$9.14 \times 10^{-2}\backslash 1.19$	$8.80 \times 10^{-2}\backslash 5.17 \times 10^{-1}$	$1.38\backslash 1.47 \times 10^{2}$
F14	Mean	1.39	1.37	3.06	1.63	1.34
	Std	1.23	0.796	3.15	0.935	1.45
	Rank	3	2	5	4	1
	Best\Worst	$9.98 \times 10^{-1}\backslash 5.93$	$9.98 \times 10^{-1}\backslash 5.93$	$0.9.98 \times 10^{-1}\backslash 10.8$	$0.9.98 \times 10^{-1}\backslash 2.98$	$0.9.98 \times 10^{-1}\backslash 10.8$
F15	Mean	$\mathbf{3.40 \times 10^{-4}}$	3.60×10^{-4}	7.04×10^{-4}	9.44×10^{-4}	7.90×10^{-4}
	Std	$\mathbf{3.39 \times 10^{-5}}$	1.45×10^{-4}	4.57×10^{-4}	3.72×10^{-4}	2.78×10^{-4}
	Rank	**1**	2	3	5	4
	Best\Worst	$\mathbf{3.14 \times 10^{-4}\backslash 5.55 \times 10^{-4}}$	$3.08 \times 10^{-4}\backslash 1.34 \times 10^{-3}$	$3.10 \times 10^{-4}\backslash 2.25 \times 10^{-3}$	$3.63 \times 10^{-4}\backslash 1.59 \times 10^{-3}$	$3.19 \times 10^{-4}\backslash 1.62 \times 10^{-3}$
F16	Mean	$\mathbf{-1.03}$	-1.03	-1.03	-1.03	-1.03
	Std	$\mathbf{1.35 \times 10^{-15}}$	1.98×10^{-8}	2.4×10^{-8}	2.16×10^{-5}	2.6×10^{-6}
	Rank	**1**	2	3	5	4
	Best\Worst	$\mathbf{-1.03\backslash -1.03}$	$-1.03\backslash -1.03$	$-1.03\backslash -1.03$	$-1.03\backslash -1.03$	$-1.03\backslash -1.03$
F17	Mean	$\mathbf{3.98 \times 10^{-1}}$	$0.3.98 \times 10^{-1}$	0.398×10^{-1}	0.399×10^{-1}	0.398×10^{-1}
	Std	$\mathbf{1.88 \times 10^{-8}}$	2.62×10^{-5}	2.15×10^{-5}	7.21×10^{-4}	1.86×10^{-5}
	Rank	**1**	1	1	1	1
	Best\Worst	$\mathbf{3.98 \times 10^{-1}\backslash 3.98 \times 10^{-1}}$	$3.98 \times 10^{-1}\backslash 3.98 \times 10^{-1}$	$3.98 \times 10^{-1}\backslash 3.98 \times 10^{-1}$	$3.98 \times 10^{-1}\backslash 0.401$	$3.98 \times 10^{-1}\backslash 3.98 \times 10^{-1}$
F18	Mean	**3.00**	3.00	3.00	3.00	3.00
	Std	$\mathbf{1.15 \times 10^{-7}}$	1.01×10^{-6}	1.62×10^{-4}	5.88×10^{-5}	2.09×10^{-4}
	Rank	**1**	1	1	1	1
	Best\Worst	$\mathbf{3.00\backslash 3.00}$	$3.00\backslash 3.00$	$3.00\backslash 3.00$	$3.00\backslash 3.00$	$3.00\backslash 3.00$
F19	Mean	$\mathbf{-3.86}$	-3.86	-3.85	-3.85	-3.86
	Std	$\mathbf{1.83 \times 10^{-3}}$	3.29×10^{-3}	1.71×10^{-2}	2.18×10^{-3}	1.01×10^{-2}
	Rank	**1**	1	1	1	1
	Best\Worst	$\mathbf{-3.86\backslash -3.86}$	$-3.86\backslash -3.85$	$-3.86\backslash -3.75$	$-3.86\backslash -3.85$	$-3.86\backslash -3.8$
F20	Mean	-3.13	-3.07	-3.26	-3.01	-3.24
	Std	1.05×10^{-1}	1.41×10^{-1}	8.66×10^{-2}	1.27×10^{-1}	7.31×10^{-2}
	Rank	3	4	1	5	2
	Best\Worst	$\mathbf{-3.31\backslash -2.74}$	$-3.30\backslash -2.73$	$-3.32\backslash -3.04$	$-3.19\backslash -2.59$	$-3.32\backslash -3.02$
F21	Mean	$\mathbf{-9.65}$	-5.35	-8.24	-3.02	-7.64
	Std	**1.38**	1.18	2.54	1.83	3.02

Table 6. *Cont.*

Benchmark		CSHHO	HHO	WOA	SCA	CSO
	Rank	**1**	4	2	5	3
	Best\Worst	**−10.2\−5.02**	−10.1\−5.04	−10.2\−2.63	−5.82\−0.497	−10.2\−2.55
F22	Mean	**−9.45**	−5.46	−7.83	−3.59	−8.19
	Std	**1.94**	1.29	\.97	1.79	2.98
	Rank	**1**	4	3	5	2
	Best\Worst	**−10.4\−5.03**	−10.1\−5.03	−10.4\−2.76	−8.23\−0.906	−10.4\−2.74
F23	Mean	**−9.97**	−5.04	−7.59	−4.1	−8.05
	Std	**1.46**	1.02	3.29	1.59	3.28
	Rank	**1**	4	3	5	2
	Best\Worst	**−10.5\−5.09**	−9.96\−1.65	−10.5\−1.67	−9.42\−0.945	$^{-}$10.5\−2.37

Table 7. The Bonferroni-Holm corrected *p*-values of Wilcoxon's signed-rank test with classic meta-heuristic algorithms.

Benchmark	CSHHO	HHO		WOA		SCA		CSO	
		Corrected *p*-Value	+\=\−	Corrected *p*-Value	+\=\−	Corrected *p*-Value	+\=\−	Corrected *p*-Value	+\=\−
F1	N/A	6.80×10^{-8}	0\0\50	6.50×10^{-8}	0\0\50	5.89×10^{-8}	0\0\50	4.84×10^{-8}	0\0\50
F2	N/A	6.73×10^{-8}	0\0\50	6.42×10^{-8}	0\0\50	5.82×10^{-8}	0\0\50	4.76×10^{-8}	0\0\50
F3	N/A	1.38×10^{-6}	5\0\45	6.35×10^{-8}	0\0\50	5.74×10^{-8}	0\0\50	4.69×10^{-8}	0\0\50
F4	N/A	6.65×10^{-8}	0\0\50	6.27×10^{-8}	0\0\50	5.67×10^{-8}	0\0\50	4.61×10^{-8}	0\0\50
F5	N/A	5.24×10^{-1}	18\0\32	6.20×10^{-8}	0\0\50	5.59×10^{-8}	0\0\50	4.53×10^{-8}	0\0\50
F6	N/A	6.57×10^{-8}	0\0\50	6.12×10^{-8}	0\0\50	5.52×10^{-8}	0\0\50	4.46×10^{-8}	0\0\50
F7	N/A	4.11×10^{-4}	11\0\39	4.27×10^{-8}	1\0\49	4.18×10^{-8}	1\0\49	4.38×10^{-8}	0\0\50
F8	N/A	1.22×10^{-3}	35\0\15	1.77×10^{-6}	8\0\42	5.44×10^{-8}	0\0\50	4.31×10^{-8}	0\0\50
F9	N/A	3.00×10^{0}	0\50\0	4.00×10^{0}	0\48\2	6.92×10^{-7}	0\8\42	8.13×10^{-1}	0\45\5
F10	N/A	3.00×10^{0}	0\50\0	3.17×10^{-6}	0\15\35	5.37×10^{-8}	0\0\50	4.23×10^{-8}	0\0\50
F11	N/A	2.00×10^{0}	0\50\0	8.13×10^{-1}	0\45\5	1.10×10^{-7}	0\3\47	3.17×10^{-3}	0\36\14
F12	N/A	6.92×10^{-7}	5\0\45	6.05×10^{-8}	0\0\50	5.29×10^{-8}	0\0\50	4.16×10^{-8}	0\0\50
F13	N/A	1.44×10^{-3}	11\0\39	5.97×10^{-8}	0\0\50	5.21×10^{-8}	0\0\50	4.08×10^{-8}	0\0\50
F14	N/A	3.93×10^{0}	6\30\14	1.15×10^{-2}	6\16\28	4.60×10^{-3}	6\2\42	3.93×10^{0}	5\38\7
F15	N/A	3.57×10^{0}	26\0\24	1.81×10^{-7}	5\0\45	5.14×10^{-8}	0\0\50	4.18×10^{-8}	1\0\49
F16	N/A	4.00×10^{0}	0\48\2	2.25×10^{0}	0\47\3	6.94×10^{-8}	0\0\50	1.15×10^{-2}	0\38\12
F17	N/A	3.58×10^{-5}	1\17\32	2.72×10^{-6}	1\11\38	5.06×10^{-8}	0\0\50	3.09×10^{-2}	3\32\15
F18	N/A	2.96×10^{-1}	2\36\12	6.87×10^{-8}	0\0\50	6.94×10^{-8}	0\0\50	5.00×10^{-1}	2\41\7
F19	N/A	3.02×10^{-2}	13\0\37	1.16×10^{-4}	10\0\40	5.33×10^{-8}	1\0\49	6.77×10^{-1}	21\0\29
F20	N/A	2.96×10^{-1}	19\0\31	1.04×10^{-5}	41\0\9	2.34×10^{-4}	13\0\37	5.59×10^{-5}	38\0\12
F21	N/A	2.23×10^{-7}	3\0\47	1.10×10^{0}	22\0\28	4.99×10^{-8}	0\0\50	2.49×10^{-4}	10\0\40
F22	N/A	2.59×10^{-7}	7\0\43	1.54×10^{0}	25\0\25	5.33×10^{-8}	1\0\49	9.79×10^{-2}	16\0\34
F23	N/A	5.43×10^{-8}	2\0\48	1.02×10^{-1}	18\0\32	4.91×10^{-8}	0\0\50	4.73×10^{-4}	14\0\36

Table 8. Average rankings obtained by classic meta-heuristic in the Friedman test, and the best result is shown in boldface.

Chaotic Mapping	Average Rankings
CSHHO	**2.57**
HHO	3.67
WOA	4.74
SCA	5.96
CSO	5.00

Figure 4 shows the convergence curves of CSHHO and the traditional optimization algorithms HHO, WOA, SCA and CSO under 23 Benchmark functions, including the performance under UM functions (F1–F7), MM functions (F8–F23), MM functions (F8–F23). The function error value was defined as, where $F(x)$ was the mean value found at all of iterations, and *F(x*)* was the optimal value recorded in 23 benchmark functions. Among them, under UM functions (F1–F7), the CSHHO algorithm converged with higher accuracy and converged faster than other algorithms, indicating that the development performance of CSHHO algorithm was improved compared with other algorithms; under MM functions (F8–F23), the CSHHO algorithm converged with higher accuracy and converged faster than other algorithms, indicating that the development performance of CSHHO algorithm was improved compared with other algorithms. From F8–F23 CSHHO algorithm did not fall into the local optimum region and could not escape; in F9, F11, F16 CSHHO converged faster in F12, F13, F17, F19, F21–F23. Although the convergence curve of the CSHHO algorithm was smoother and converged slower in the initial iterations as the algorithm searched further. In F10, F15 the CSHHO algorithm not only explores the dominant region with faster convergence speed, but also leads the rest of the algorithms in terms of search accuracy. Therefore, CSHHO algorithm benefits from Gauss chaotic mapping that enhances the population initialization of the algorithm, as well as adaptive weighting mechanism, variable spiral position updating mechanism and adaptive neighborhood disturbance mechanism that enhance the exploration and exploitation ability of the algorithm, CSHHO is less likely to fall into the current search region and increase the ability to jump out of the local optimal region.

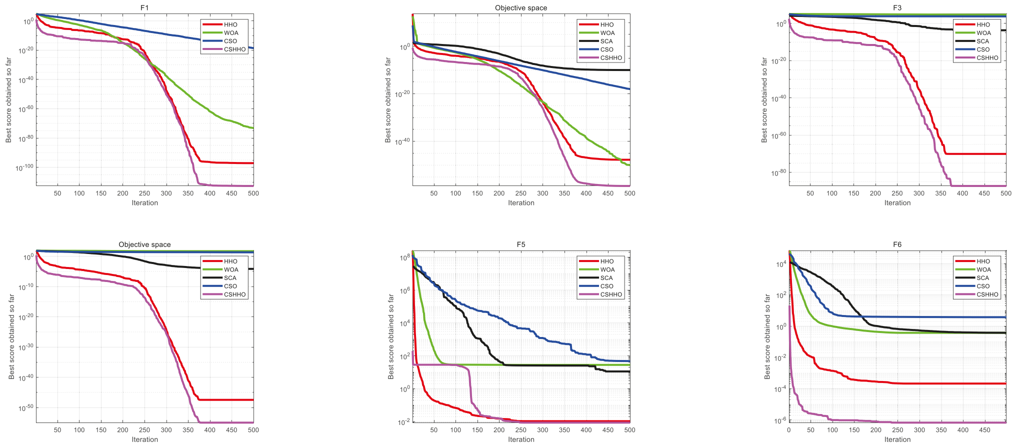


Figure 4. *Cont.*

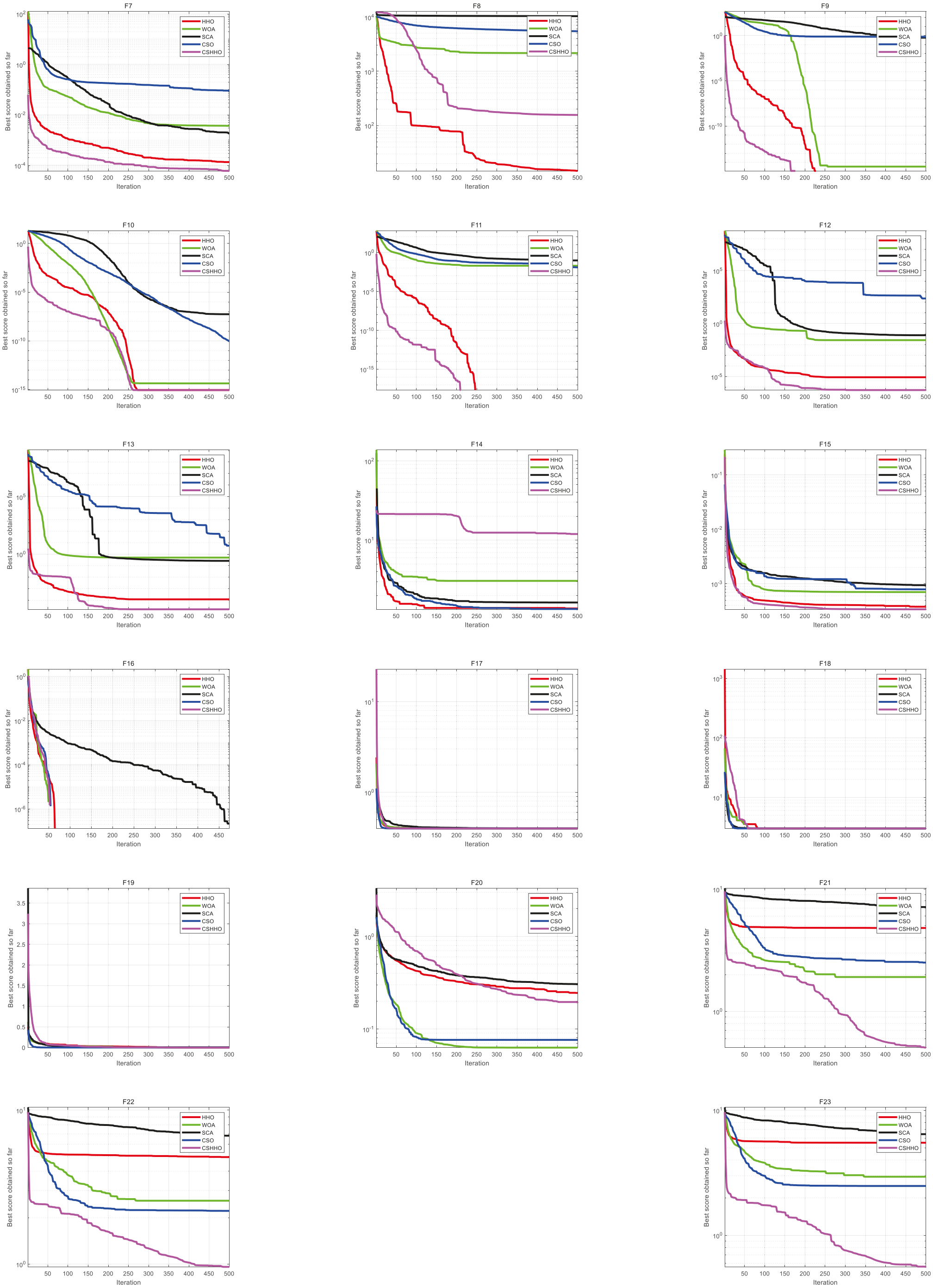


Figure 4. Convergence curves of CSHHO and the classic meta-heuristic algorithms.

4.2.3. Comparison with HHO Variants

In order to verify the effectiveness of the CSHHO algorithm against HHO variants in recent years, this subsection compares the CSHHO algorithm with the recently published advanced HHO variants: GCHHO and DEPSOASS and Improved GSA and DGOBLFOA, observing the mean accuracy mean and stability Std of each algorithm. Table 9 presents the results of the experiments. Next, using nonparametric tests: the Wilcoxon signed-rank test and the Friedman test were used to synthetically assess the performance differences between CSHHO and GCHHO and DEPSOASS and Improved GSA and DGOBLFOA. The configuration of the experiments is the same as in Section 4.2.2, "$+\backslash=\backslash-$": the number of CSHHO results obtained from 50 runs in each test function that are worse, similar, or better than the comparison algorithm. Table 10 presents the results of the Bonferroni–Holm correction of Wilcoxon signed-rank test experiments, Table 11 presents the results of the Friedman test experiments.

Analyzing the data in Table 9, CSHHO has some advantages over the advanced HHO variants: under the UM functions (F1–F7), CSHHO outperforms the other algorithms in F1–F4, F6, F7, which indicates that CSHHO has further enhanced global best-finding capability compared to the advanced HHO variants. Under the MM functions (F8–F23) the CSHHO's performance outperforms the remaining algorithms in F9, F10, F11, F16, F17, F18, F19, F21, which indicates that CSHHO can explore the peak-to-peak information deeply and effectively to avoid the algorithm from entering the local optimum. In summary, CSHHO has a good ability to develop and explore and avoid local optima.

Table 10 shows the results of the corrected Wilcoxon signed rank test at 5% confidence level and Friedman's test for CSHHO and GCHHO and DEPSOASS and Improved GSA and DGOBLFOA. The p-values indicate whether the numerical results of the algorithms involved in the comparison are significant compared to CSHHO; if the p-value is greater than 0.05, the numerical results of the algorithm are considered the same as CSHHO; otherwise, it is considered to be significantly different. Analyzing the Bonferroni–Holm corrected p-value column values in Table 10, under the 23 classical test functions, only GCHHO is less different from CSHHO in the F9–F11 and F16–18 test functions, and in the rest of the test functions. Moreover, at Table 11 upon analyzing the results of the Friedman test, the value of CSHHO is 1.70, which is lower than others. The result indicates that CSHHO has an advantage over the above algorithms in optimization.

Figure 5 shows the convergence curves of CSHHO and the variants of the optimization algorithm GCHHO and DEPSOASS and Improved GSA and DGOBLFOA under 23 Benchmark Functions, including the performance under UM functions (F1–F7), MM functions (F8–F23). The function error value is defined as, where $F(x)$ is the mean value found at all of iterations, and $F(x^*)$ is the optimal value is recorded in 23 The CSHHO algorithm under UM functions (F1–F7) has improved convergence accuracy in F1–F4, F6, F7 compared to other algorithms, and convergence speed is better than other algorithms in F1, F2, F3, F4, F5, F6, F7; under MM functions (F8–F23), CSHHO algorithm does not fall into the local optimal region and cannot escape, in F9, F10, F11, F16, F17, F18, F19, F21. CSHHO can explore the dominant region well and is ahead of other algorithms in terms of search accuracy. In F9, F10, F11, F12, F13, F21, F22, F23, CSHHO has smooth convergence curves and faster convergence speed. In F16–F20, although the convergence speed of the CSHHO algorithm is slow, the convergence curve does not produce large fluctuations, which indicates that the CSHHO algorithm has good search ability, and it does not fall into local optimum and cannot jump out. Functions. In summary, the CSHHO algorithm's performance is further improved compared to GCHHO algorithm and other advanced algorithms. Compared to these advanced variants, CSHHO algorithm are effective.

Table 9. Results of a comparison with the advanced meta-heuristic algorithms.

Benchmark		CSHHO	GCHHO	DEPSOASS	Improved GSA	DGOBLFOA
F1	Mean	**1.96×10^{-113}**	3.91×10^{-97}	1.45×10^{-3}	2.84×10^{1}	2.88×10^{-8}
	Std	**1.38×10^{-112}**	2.55×10^{-96}	1.45×10^{-3}	2.84×10^{1}	2.88×10^{-8}
	Rank	**1**	2	4	5	3
	Best\Worst	**$9.78 \times 10^{-112}\backslash 1.44 \times 10^{-138}$**	$1.80 \times 10^{-95}\backslash 1.25 \times 10^{-119}$	$3.22 \times 10^{-3}\backslash 3.86 \times 10^{-4}$	$4.68 \times 10^{1}\backslash 1.51 \times 10^{1}$	$1.42 \times 10^{-6}\backslash 0.00 \times 10^{0}$
F2	Mean	**1.66×10^{-59}**	1.62×10^{-50}	1.88×10^{-1}	2.81×10^{1}	2.54×10^{-1}
	Std	**1.06×10^{-58}**	9.45×10^{-50}	1.88×10^{-1}	2.81×10^{1}	2.54×10^{-1}
	Rank	**1**	2	3	5	4
	Best\Worst	**$7.47 \times 10^{-58}\backslash 8.54 \times 10^{-70}$**	$6.63 \times 10^{-49}\backslash 3.57 \times 10^{-60}$	$2.84 \times 10^{-1}\backslash 1.19 \times 10^{-1}$	$3.46 \times 10^{1}\backslash 1.44 \times 10^{1}$	$1.03 \times 10^{0}\backslash 8.72 \times 10^{-4}$
F3	Mean	**4.96×10^{-88}**	1.46×10^{-65}	5.37×10^{-1}	3.32×10^{2}	1.18×10^{-2}
	Std	**2.81×10^{-87}**	1.01×10^{-64}	5.37×10^{-1}	3.32×10^{2}	1.18×10^{-2}
	Rank	**1**	2	4	5	3
	Best\Worst	**$1.96 \times 10^{-86}\backslash 5.85 \times 10^{-108}$**	$7.16 \times 10^{-64}\backslash 1.16 \times 10^{-91}$	$1.44 \times 10^{0}\backslash 9.11 \times 10^{-2}$	$5.38 \times 10^{2}\backslash 1.39 \times 10^{2}$	$4.66 \times 10^{-1}\backslash 1.45 \times 10^{-24}$
F4	Mean	**1.05×10^{-55}**	1.82×10^{-48}	3.49×10^{-2}	2.31×10^{0}	1.74×10^{-4}
	Std	**6.13×10^{-55}**	9.06×10^{-48}	3.49×10^{-2}	2.31×10^{0}	1.74×10^{-4}
	Rank	**1**	2	4	5	3
	Best\Worst	**$4.25 \times 10^{-54}\backslash 1.83 \times 10^{-68}$**	$5.94 \times 10^{-47}\backslash 2.66 \times 10^{-58}$	$5.40 \times 10^{-2}\backslash 1.75 \times 10^{-2}$	$2.83 \times 10^{0}\backslash 1.44 \times 10^{0}$	$8.57 \times 10^{-3}\backslash 1.29 \times 10^{-20}$
F5	Mean	8.57×10^{-3}	7.14×10^{-4}	2.76×10^{1}	1.19×10^{4}	2.87×10^{1}
	Std	2.23×10^{-2}	7.02×10^{-4}	2.76×10^{1}	1.19×10^{4}	2.87×10^{1}
	Rank	2	1	3	5	4
	Best\Worst	$1.32 \times 10^{-1}\backslash 1.29 \times 10^{-8}$	$3.28 \times 10^{-3}\backslash 2.96 \times 10^{-5}$	$2.95 \times 10^{1}\backslash 2.59 \times 10^{1}$	$2.60 \times 10^{4}\backslash 1.29 \times 10^{3}$	$3.20 \times 10^{1}\backslash 2.79 \times 10^{1}$
F6	Mean	**6.73×10^{-7}**	1.20×10^{-6}	1.30×10^{-3}	2.83×10^{1}	5.35×10^{0}
	Std	**1.85×10^{-6}**	1.61×10^{-6}	1.30×10^{-3}	2.83×10^{1}	5.35×10^{0}
	Rank	**1**	2	3	5	4
	Best\Worst	**$8.80 \times 10^{-6}\backslash 4.40 \times 10^{-14}$**	$1.02 \times 10^{-5}\backslash 1.47 \times 10^{-7}$	$3.27 \times 10^{-3}\backslash 3.91 \times 10^{-4}$	$4.15 \times 10^{1}\backslash 1.43 \times 10^{1}$	$6.03 \times 10^{0}\backslash 4.32 \times 10^{0}$
F7	Mean	**5.98×10^{-5}**	1.94×10^{-4}	1.26×10^{-1}	1.04×10^{2}	6.10×10^{-1}
	Std	**7.15×10^{-5}**	1.92×10^{-4}	1.26×10^{-1}	1.04×10^{2}	6.10×10^{-1}
	Rank	**1**	2	3	5	4
	Best\Worst	**$4.65 \times 10^{-4}\backslash 2.03 \times 10^{-7}$**	$7.84 \times 10^{-4}\backslash 5.01 \times 10^{-6}$	$2.21 \times 10^{-1}\backslash 7.33 \times 10^{-2}$	$1.52 \times 10^{2}\backslash 3.44 \times 10^{0}$	$1.44 \times 10^{0}\backslash 1.01 \times 10^{-1}$
F8	Mean	-1.24×10^{4}	-1.25×10^{4}	-2.78×10^{3}	-2.70×10^{3}	-3.70×10^{2}
	Std	2.71×10^{2}	1.67×10^{2}	-2.78×10^{3}	-2.70×10^{3}	1.69×10^{2}
	Rank	2	1	3	4	5
	Best\Worst	$-1.15 \times 10^{4}\backslash -1.26 \times 10^{4}$	$-1.18 \times 10^{4}\backslash -1.26 \times 10^{4}$	$-1.79 \times 10^{3}\backslash -3.79 \times 10^{3}$	$-1.74 \times 10^{3}\backslash -3.42 \times 10^{3}$	$-1.79 \times 10^{3}\backslash -3.46 \times 10^{3}$
F9	Mean	**0.00**	0.00	3.72×10^{1}	2.54×10^{2}	8.95×10^{0}
	Std	**0.00**	0.00	3.72×10^{1}	2.54×10^{2}	8.95×10^{0}
	Rank	**1**	1	4	5	3
	Best\Worst	**0.00\0.00**	0.00\0.00	$6.02 \times 10^{1}\backslash 2.00 \times 10^{1}$	$2.91 \times 10^{2}\backslash 1.96 \times 10^{2}$	$2.81 \times 10^{1}\backslash 2.18 \times 10^{-3}$
F10	Mean	**8.88×10^{-16}**	8.88×10^{-16}	3.06×10^{-2}	5.11×10^{0}	4.23×10^{-2}
	Std	**1.99×10^{-31}**	1.99×10^{-31}	3.06×10^{-2}	5.11×10^{0}	4.23×10^{-2}
	Rank	**1**	1	3	5	4
	Best\Worst	**$8.88 \times 10^{-16}\backslash 8.88 \times 10^{-16}$**	$8.88 \times 10^{-16}\backslash 8.88 \times 10^{-16}$	$4.31 \times 10^{-2}\backslash 1.67 \times 10^{-2}$	$6.09 \times 10^{0}\backslash 3.64 \times 10^{0}$	$6.08 \times 10^{-1}\backslash 8.88 \times 10^{-16}$
F11	Mean	**0.00×10^{0}**	0.00×10^{0}	5.90×10^{-4}	7.85×10^{-1}	1.09×10^{-9}

Table 9. *Cont.*

Benchmark		CSHHO	GCHHO	DEPSOASS	Improved GSA	DGOBLFOA
	Std	**0.00×10^{0}**	0.00×10^{0}	5.90×10^{-4}	7.85×10^{-1}	1.09×10^{-9}
	Rank	**1**	1	4	5	3
F12	Best\Worst	**$0.00 \backslash 0.00$**	$0.00 \backslash 0.00$	$1.97 \times 10^{-2} \backslash 2.27 \times 10^{-5}$	$9.59 \times 10^{-1} \backslash 4.58 \times 10^{-1}$	$5.46 \times 10^{-8} \backslash 0.00 \times 10^{0}$
	Mean	5.26×10^{-7}	8.05×10^{-8}	9.89×10^{-6}	1.29×10^{0}	6.98×10^{-1}
	Std	1.20×10^{-6}	9.54×10^{-8}	9.89×10^{-6}	1.29×10^{0}	6.98×10^{-1}
	Rank	2	1	3	5	4
F13	Best\Worst	$5.00 \times 10^{-6} \backslash 1.68 \times 10^{-11}$	$5.02 \times 10^{-7} \backslash 5.77 \times 10^{-9}$	$2.58 \times 10^{-5} \backslash 4.76 \times 10^{-6}$	$2.32 \times 10^{0} \backslash 6.10 \times 10^{-1}$	$1.01 \times 10^{0} \backslash 3.67 \times 10^{-1}$
	Mean	1.64×10^{-5}	1.26×10^{-6}	1.69×10^{-3}	6.34×10^{0}	2.44×10^{0}
	Std	2.86×10^{-5}	1.48×10^{-6}	1.69×10^{-3}	6.34×10^{0}	2.44×10^{0}
	Rank	2	1	3	5	4
F14	Best\Worst	$1.23 \times 10^{-4} \backslash 6.14 \times 10^{-10}$	$6.48 \times 10^{-6} \backslash 4.00 \times 10^{-8}$	$1.12 \times 10^{-2} \backslash 6.80 \times 10^{-5}$	$9.67 \times 10^{0} \backslash 2.90 \times 10^{0}$	$2.88 \times 10^{0} \backslash 1.60 \times 10^{0}$
	Mean	1.39×10^{0}	9.98×10^{-1}	1.52×10^{0}	7.07×10^{0}	4.34×10^{0}
	Std	1.23×10^{0}	5.61×10^{-16}	1.52×10^{0}	7.07×10^{0}	4.34×10^{0}
	Rank	2	1	3	5	4
F15	Best\Worst	$5.93 \times 10^{0} \backslash 9.98 \times 10^{-1}$	$9.98 \times 10^{-1} \backslash 9.98 \times 10^{-1}$	$3.02 \times 10^{0} \backslash 9.98 \times 10^{-1}$	$1.55 \times 10^{1} \backslash 1.01 \times 10^{0}$	$1.18 \times 10^{1} \backslash 9.98 \times 10^{-1}$
	Mean	3.40×10^{-4}	3.17×10^{-4}	9.62×10^{-4}	1.39×10^{-3}	2.61×10^{-2}
	Std	3.39×10^{-5}	4.50×10^{-5}	9.62×10^{-4}	1.39×10^{-3}	2.61×10^{-2}
	Rank	2	1	3	4	5
F16	Best\Worst	**$5.55 \times 10^{-4} \backslash 3.14 \times 10^{-4}$**	$5.41 \times 10^{-4} \backslash 3.07 \times 10^{-4}$	$4.55 \times 10^{-3} \backslash 5.50 \times 10^{-4}$	$3.18 \times 10^{-3} \backslash 1.03 \times 10^{-3}$	$6.88 \times 10^{-2} \backslash 1.81 \times 10^{-3}$
	Mean	**-1.03×10^{0}**	-1.03×10^{0}	-1.03×10^{0}	-1.02×10^{0}	-1.03×10^{0}
	Std	**1.35×10^{-15}**	1.35×10^{-15}	-1.03×10^{0}	-1.02×10^{0}	-1.03×10^{0}
	Rank	**1**	1	1	5	1
F17	Best\Worst	**$-1.03 \times 10^{0} \backslash -1.03 \times 10^{0}$**	$-1.03 \times 10^{0} \backslash -1.03 \times 10^{0}$	$-1.03 \times 10^{0} \backslash -1.03 \times 10^{0}$	$-9.22 \times 10^{-1} \backslash -1.03 \times 10^{0}$	$-1.03 \times 10^{0} \backslash -1.03 \times 10^{0}$
	Mean	**3.98×10^{-1}**	3.98×10^{-1}	3.98×10^{-1}	4.08×10^{-1}	5.30×10^{-1}
	Std	**1.88×10^{-8}**	1.12×10^{-16}	3.98×10^{-1}	4.08×10^{-1}	5.30×10^{-1}
	Rank	**1**	1	1	4	5
F18	Best\Worst	**$3.98 \times 10^{-13} \backslash 98 \times 10^{-1}$**	$3.98 \times 10^{-1} \backslash 3.98 \times 10^{-1}$	$3.98 \times 10^{-1} \backslash 3.98 \times 10^{-1}$	$4.35 \times 10^{-1} \backslash 3.98 \times 10^{-1}$	$1.70 \times 10^{0} \backslash 4.00 \times 10^{-1}$
	Mean	**3.00×10^{0}**	3.00×10^{0}	3.00×10^{0}	4.91×10^{0}	3.50×10^{0}
	Std	**1.15×10^{-7}**	0.00×10^{0}	3.00×10^{0}	4.91×10^{0}	3.50×10^{0}
	Rank	**1**	1	1	5	4
F19	Best\Worst	**$3.00 \times 10^{0} \backslash 3.00 \times 10^{0}$**	$3.00 \times 10^{0} \backslash 3.00 \times 10^{0}$	$3.00 \times 10^{0} \backslash 3.00 \times 10^{0}$	$1.25 \times 10^{1} \backslash 3.01 \times 10^{0}$	$6.51 \times 10^{0} \backslash 3.01 \times 10^{0}$
	Mean	**-3.86×10^{0}**	-3.86×10^{0}	-3.86×10^{0}	-3.45×10^{0}	-3.50×10^{0}
	Std	**1.83×10^{-3}**	2.24×10^{-15}	-3.86×10^{0}	-3.45×10^{0}	-3.50×10^{0}
	Rank	**1**	1	1	5	4
F20	Best\Worst	**$-3.86 \times 10^{0} \backslash -3.86 \times 10^{0}$**	$-3.86 \times 10^{0} \backslash -3.86 \times 10^{0}$	$-3.86 \times 10^{0} \backslash -3.86 \times 10^{0}$	$-2.90 \times 10^{0} \backslash -3.86 \times 10^{0}$	$-2.69 \times 10^{0} \backslash -3.85 \times 10^{0}$
	Mean	-3.13×10^{0}	-3.25×10^{0}	-3.31×10^{0}	-1.69×10^{0}	-2.09×10^{0}
	Std	1.05×10^{-1}	5.88×10^{-2}	-3.31×10^{0}	-1.69×10^{0}	-2.09×10^{0}
	Rank	3	2	1	5	4
F21	Best\Worst	$-2.74 \times 10^{0} \backslash -3.31 \times 10^{0}$	$-3.20 \times 10^{0} \backslash -3.32 \times 10^{0}$	$-3.18 \times 10^{0} \backslash -3.32 \times 10^{0}$	$-6.72 \times 10^{-1} \backslash -2.55 \times 10^{0}$	$-1.11 \times 10^{0} \backslash -3.10 \times 10^{0}$
	Mean	**-9.65×10^{0}**	-6.07×10^{0}	-6.85×10^{0}	-4.05×10^{0}	-4.49×10^{0}
	Std	**1.38×10^{0}**	2.06×10^{0}	-6.85×10^{0}	-4.05×10^{0}	-4.49×10^{0}

Table 9. *Cont.*

Benchmark		CSHHO	GCHHO	DEPSOASS	Improved GSA	DGOBLFOA
	Rank	**1**	3	2	5	4
	Best\Worst	**$-5.02 \times 10^{0}\backslash-1.02 \times 10^{1}$**	$-5.06 \times 10^{0}\backslash-1.02 \times 10^{1}$	$-2.68 \times 10^{0}\backslash-1.02 \times 10^{1}$	$-1.99 \times 10^{0}\backslash-7.93 \times 10^{0}$	$-2.67 \times 10^{0}\backslash-7.66 \times 10^{0}$
F22	Mean	-9.45×10^{0}	-5.94×10^{0}	-1.01×10^{1}	-3.55×10^{0}	-4.08×10^{0}
	Std	1.94×10^{0}	1.97×10^{0}	-1.01×10^{1}	-3.55×10^{0}	-4.08×10^{0}
	Rank	2	3	1	5	4
	Best\Worst	$-5.03 \times 10^{0}\backslash-1.04 \times 10^{1}$	$-5.09 \times 10^{0}\backslash-1.04 \times 10^{1}$	$-6.48 \times 10^{0}\backslash-1.04 \times 10^{1}$	$-1.93 \times 10^{0}\backslash-7.27 \times 10^{0}$	$-1.67 \times 10^{0}\backslash-8.98 \times 10^{0}$
F23	Mean	-9.97×10^{0}	-5.67×10^{0}	-1.03×10^{1}	-4.34×10^{0}	-4.48×10^{0}
	Std	1.46×10^{0}	1.64×10^{0}	-1.03×10^{1}	-4.34×10^{0}	-4.48×10^{0}
	Rank	2	3	1	5	4
	Best\Worst	$-5.09 \times 10^{0}\backslash-1.05 \times 10^{1}$	$-5.13 \times 10^{0}\backslash-1.05 \times 10^{1}$	$-5.42 \times 10^{0}\backslash-1.05 \times 10^{1}$	$-2.61 \times 10^{0}\backslash-7.03 \times 10^{0}$	$-2.43 \times 10^{0}\backslash-9.12 \times 10^{0}$

Table 10. The Bonferroni-Holm corrected *p*-values of Wilcoxon's signed-rank test with the advanced meta-heuristic algorithms.

Benchmark		GCHHO		DEPSOASS		GSA		FOA	
		Corrected *p*-Value	+\=\−	Corrected *p*-Value	+\=\−	Corrected *p*-Value	+\=\−	Corrected *p*-Value	+\=\−
F1	N/A	3.07×10^{-8}	1\0\49	6.57×10^{-8}	0\0\50	5.06×10^{-8}	0\0\50	2.99×10^{-8}	2\0\48
F2	N/A	6.80×10^{-8}	0\0\50	6.50×10^{-8}	0\0\50	4.99×10^{-8}	0\0\50	4.01×10^{-8}	0\0\50
F3	N/A	6.73×10^{-8}	0\0\50	6.42×10^{-8}	0\0\50	4.91×10^{-8}	0\0\50	3.93×10^{-8}	0\0\50
F4	N/A	6.65×10^{-8}	0\0\50	6.35×10^{-8}	0\0\50	4.84×10^{-8}	0\0\50	3.85×10^{-8}	0\0\50
F5	N/A	3.46×10^{-1}	25\0\25	6.27×10^{-8}	0\0\50	4.76×10^{-8}	0\0\50	3.78×10^{-8}	0\0\50
F6	N/A	4.84×10^{-3}	9\0\41	6.20×10^{-8}	0\0\50	4.69×10^{-8}	0\0\50	3.70×10^{-8}	0\0\50
F7	N/A	4.72×10^{-4}	12\0\38	6.12×10^{-8}	0\0\50	4.61×10^{-8}	0\0\50	3.63×10^{-8}	0\0\50
F8	N/A	3.70×10^{-1}	35\0\15	6.05×10^{-8}	0\0\50	4.53×10^{-8}	0\0\50	3.55×10^{-8}	50\0\0
F9	N/A	4.00×10^{0}	0\50\0	5.97×10^{-8}	0\0\50	4.46×10^{-8}	0\0\50	3.48×10^{-8}	0\0\50
F10	N/A	4.00×10^{0}	0\50\0	5.89×10^{-8}	0\0\50	4.38×10^{-8}	0\0\50	4.71×10^{-8}	0\2\48
F11	N/A	3.00×10^{0}	0\50\0	5.82×10^{-8}	0\0\50	4.31×10^{-8}	0\0\50	8.59×10^{-2}	0\42\8
F12	N/A	8.60×10^{-1}	26\0\24	5.74×10^{-8}	0\0\50	4.23×10^{-8}	0\0\50	3.40×10^{-8}	0\0\50
F13	N/A	7.25×10^{-4}	36\0\14	5.67×10^{-8}	0\0\50	4.16×10^{-8}	0\0\50	3.33×10^{-8}	0\0\50
F14	N/A	3.13×10^{-1}	6\44\0	1.02×10^{-7}	2\0\48	1.20×10^{-3}	5\1\44	1.39×10^{-7}	1\0\49
F15	N/A	3.99×10^{-6}	48\0\2	5.59×10^{-8}	0\0\50	4.08×10^{-8}	0\0\50	3.25×10^{-8}	0\0\50
F16	N/A	2.00×10^{0}	0\50\0	5.52×10^{-8}	0\0\50	6.75×10^{-8}	0\0\50	3.17×10^{-4}	0\26\24
F17	N/A	8.75×10^{-1}	4\46\0	5.44×10^{-8}	0\0\50	4.71×10^{-8}	0\2\48	3.17×10^{-8}	0\0\50
F18	N/A	8.75×10^{-1}	4\46\0	5.37×10^{-8}	0\0\50	6.88×10^{-8}	0\0\50	3.10×10^{-8}	0\0\50
F19	N/A	6.88×10^{-8}	50\0\0	5.29×10^{-8}	0\0\50	3.04×10^{-7}	43\0\7	3.02×10^{-8}	0\0\50
F20	N/A	1.10×10^{-6}	45\0\5	5.21×10^{-8}	0\0\50	2.99×10^{-8}	49\0\1	3.08×10^{-8}	1\0\49
F21	N/A	1.83×10^{-5}	13\0\37	3.08×10^{-8}	2\0\48	1.89×10^{-3}	19\0\31	3.26×10^{-8}	2\0\48
F22	N/A	4.72×10^{-4}	13\0\37	5.14×10^{-8}	0\0\50	1.55×10^{-3}	44\0\6	2.95×10^{-8}	0\0\50
F23	N/A	2.67×10^{-7}	9\0\41	3.07×10^{-8}	1\0\49	2.46×10^{-5}	47\0\3	2.97×10^{-8}	1\0\49

Table 11. Average rankings obtained by advanced meta-heuristic in the Friedman test, and the best result is shown in boldface.

Chaotic Mapping	Average Rankings
CSHHO	**1.70**
GCHHO	1.83
DEPSOASS	2.76
GSA	4.87
FOA	3.85

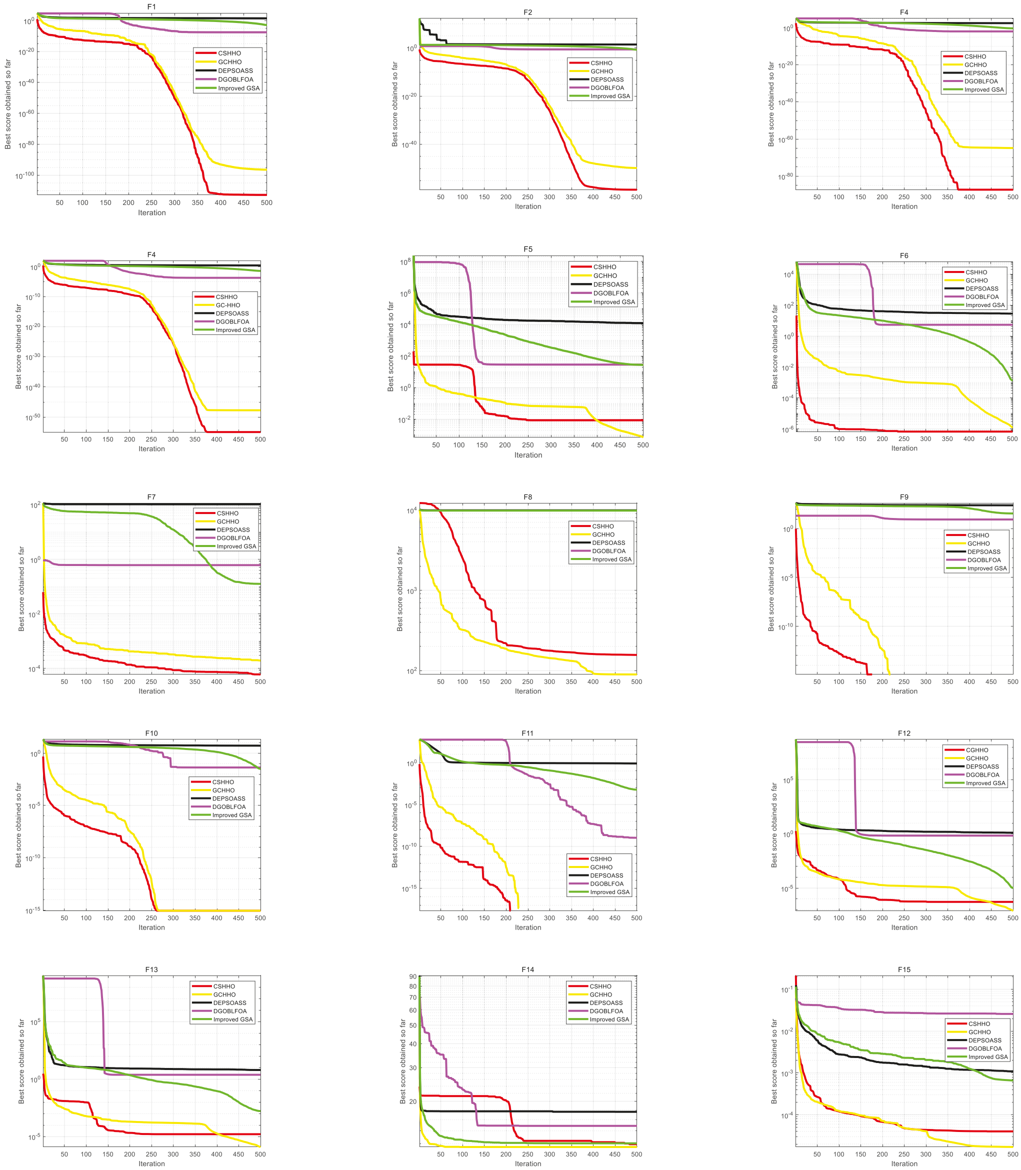


Figure 5. *Cont.*

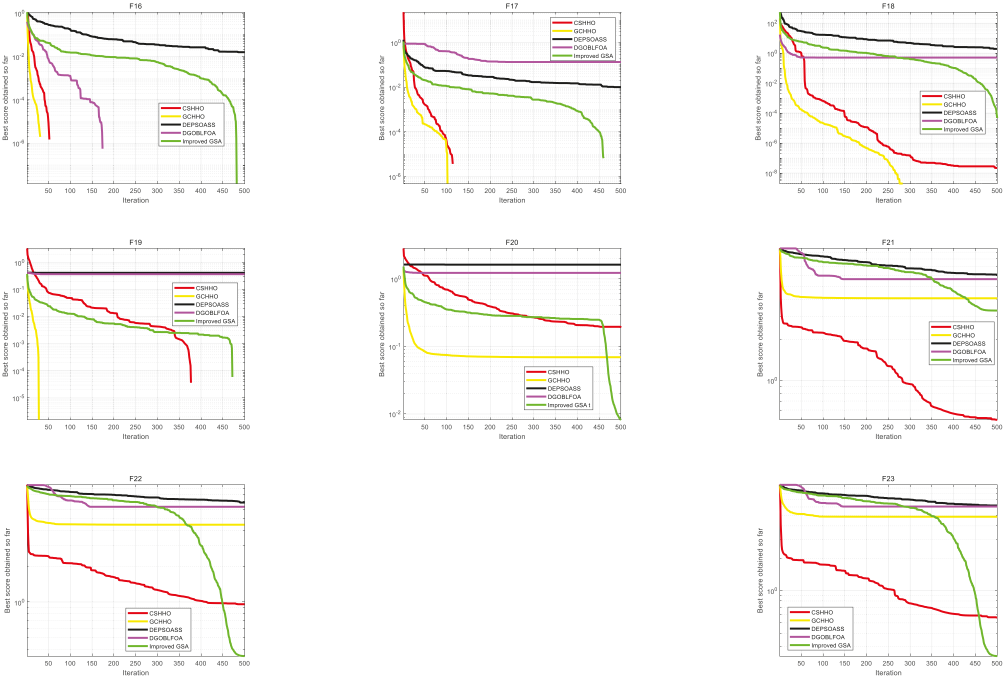


Figure 5. Convergence curves of CSHHO and the advanced meta-heuristic algorithms.

4.2.4. Scalability Test on CSHHO

Dimensional data are an important basis for analyzing the influence of the number of factors to be optimized on the algorithm, and the purpose of the scalability test is to further verify the overall performance and stability of the optimization model. The experimental subjects in this section are CSHHO, HHO. 29 CEC2017 functions [72] based on 50 and 100 dimensions, respectively, are used for scalability experiments. In this experiment, the experimental parameters and the experimental environment are consistent with the previous experiments except that the dimensionality settings are different from the previous experiments, and Table 12 shows the experimental results using Mean and Std.

The best numerical results in CSHHO and HHO are set in bold, and the numerical results of both equivalents are in bold. Under the UM function (F1–F2), the numerical results of CSHHO are overall better than those of HHO, and CSHHO continues to maintain some advantage as the number of dimensions increases; under the MM function test (F3–F9), CSHHO performs better than HHO in 50 and 100 dimensions. The CSHHO's performance in the 50 and 100 dimensions was generally better than that of HHO in the 50 dimensions in F6 and F7 and in the 100 dimensions in F4. In the hybrid function (F10–F19), CSHHO performs better in the rest of the test sets, except for F11 and F12, where it performs lower than HHO. In the composition function (F20–F29), it still has good ability, and the accuracy in functions F21, F22 and F26 is higher than HHO, and the effect in functions F22, F23, F24, F26 is the same as HHO, and the effect in functions F23 (50 dimensions), F26 (100 dimensions) is lower than HHO. In general, compared with HHO, CSHHO can better balance the exploration and exploitation process as the number of dimensions increases.

Table 12. Experimental results of scalability tests in different dimensions.

Benchmark	Metric	50		100	
		CSHHO	HHO	CSHHO	HHO
F1	Mean\Std	$3.49 \times 10^{7}\backslash 1.19 \times 10^{8}$	$3.30 \times 10^{8}\backslash 9.91 \times 10^{7}$	$1.08 \times 10^{10}\backslash 2.06 \times 10^{9}$	$1.10 \times 10^{10}\backslash 2.09 \times 10^{9}$
F2	Mean\Std	$9.88 \times 10^{4}\backslash 1.66 \times 10^{4}$	$1.03 \times 10^{5}\backslash 1.38 \times 10^{4}$	$2.78 \times 10^{5}\backslash 1.73 \times 10^{4}$	$2.90 \times 10^{5}\backslash 1.42 \times 10^{4}$
F3	Mean\Std	$8.65 \times 10^{2}\backslash 1.16 \times 10^{2}$	$9.00 \times 10^{2}\backslash 1.52 \times 10^{2}$	$2.89 \times 10^{2}\backslash 4.96 \times 10^{2}$	$3.01 \times 10^{3}\backslash 5.84 \times 10^{2}$
F4	Mean\Std	$8.95 \times 10^{2}\backslash 2.82 \times 10^{1}$	$8.96 \times 10^{2}\backslash 3.47 \times 10^{1}$	$1.56 \times 10^{3}\backslash 4.55 \times 10^{1}$	$1.56 \times 10^{3}\backslash 5.35 \times 10^{1}$
F5	Mean\Std	$6.71 \times 10^{2}\backslash 4.36 \times 10^{0}$	$6.74 \times 10^{2}\backslash 4.16 \times 10^{0}$	$6.85 \times 10^{2}\backslash 2.82 \times 10^{0}$	$6.85 \times 10^{2}\backslash 3.25 \times 10^{0}$
F6	Mean\Std	$1.84 \times 10^{3}\backslash 9.07 \times 10^{1}$	$1.84 \times 10^{3}\backslash 7.88 \times 10^{1}$	$3.71 \times 10^{3}\backslash 1.41 \times 10^{2}$	$3.69 \times 10^{3}\backslash 1.46 \times 10^{2}$
F7	Mean\Std	$1.20 \times 10^{3}\backslash 3.34 \times 10^{1}$	$1.20 \times 10^{3}\backslash 3.11 \times 10^{1}$	$2.01 \times 10^{3}\backslash 7.14 \times 10^{1}$	$2.01 \times 10^{3}\backslash 6.02 \times 10^{1}$
F8	Mean\Std	$2.52 \times 10^{4}\backslash 2.86 \times 10^{3}$	$2.54 \times 10^{4}\backslash 3.63 \times 10^{3}$	$5.24 \times 10^{4}\backslash 5.53 \times 10^{3}$	$5.72 \times 10^{4}\backslash 5.52 \times 10^{3}$
F9	Mean\Std	$9.41 \times 10^{3}\backslash 9.83 \times 10^{2}$	$9.49 \times 10^{3}\backslash 8.76 \times 10^{2}$	$2.21 \times 10^{4}\backslash 1.46 \times 10^{3}$	$2.28 \times 10^{4}\backslash 2.12 \times 10^{3}$
F10	Mean\Std	$1.65 \times 10^{3}\backslash 1.47 \times 10^{2}$	$1.66 \times 10^{3}\backslash 1.21 \times 10^{2}$	$5.03 \times 10^{4}\backslash 1.59 \times 10^{4}$	$5.06 \times 10^{4}\backslash 1.06 \times 10^{4}$
F11	Mean\Std	$2.17 \times 10^{8}\backslash 1.28 \times 10^{8}$	$1.88 \times 10^{8}\backslash 1.05 \times 10^{8}$	$1.58 \times 10^{9}\backslash 5.29 \times 10^{8}$	$1.68 \times 10^{9}\backslash 5.46 \times 10^{8}$
F12	Mean\Std	$3.82 \times 10^{6}\backslash 1.44 \times 10^{6}$	$5.20 \times 10^{6}\backslash 4.28 \times 10^{6}$	$1.75 \times 10^{7}\backslash 7.23 \times 10^{6}$	$1.62 \times 10^{7}\backslash 6.97 \times 10^{6}$
F13	Mean\Std	$1.54 \times 10^{6}\backslash 1.01 \times 10^{6}$	$2.07 \times 10^{6}\backslash 2.53 \times 10^{6}$	$4.81 \times 10^{6}\backslash 1.82 \times 10^{6}$	$5.41 \times 10^{6}\backslash 1.64 \times 10^{6}$
F14	Mean\Std	$6.00 \times 10^{5}\backslash 2.65 \times 10^{5}$	$6.35 \times 10^{5}\backslash 2.81 \times 10^{5}$	$3.36 \times 10^{6}\backslash 9.38 \times 10^{5}$	$4.35 \times 10^{6}\backslash 4.49 \times 10^{6}$
F15	Mean\Std	$4.38 \times 10^{3}\backslash 6.23 \times 10^{2}$	$4.52 \times 10^{3}\backslash 6.15 \times 10^{2}$	$8.55 \times 10^{3}\backslash 1.00 \times 10^{3}$	$8.56 \times 10^{3}\backslash 8.52 \times 10^{2}$
F16	Mean\Std	$3.78 \times 10^{3}\backslash 3.76 \times 10^{2}$	$3.90 \times 10^{3}\backslash 4.41 \times 10^{2}$	$6.73 \times 10^{3}\backslash 7.03 \times 10^{2}$	$6.84 \times 10^{3}\backslash 7.62 \times 10^{2}$
F17	Mean\Std	$5.39 \times 10^{5}\backslash 4.79 \times 10^{6}$	$4.63 \times 10^{6}\backslash 4.40 \times 10^{6}$	$5.75 \times 10^{6}\backslash 2.56 \times 10^{6}$	$6.62 \times 10^{6}\backslash 3.52 \times 10^{6}$
F18	Mean\Std	$5.82 \times 10^{5}\backslash 6.11 \times 10^{5}$	$1.12 \times 10^{6}\backslash 7.19 \times 10^{5}$	$1.20 \times 10^{7}\backslash 4.90 \times 10^{6}$	$1.55 \times 10^{7}\backslash 6.97 \times 10^{6}$
F19	Mean\Std	$3.63 \times 10^{2}\backslash 3.07 \times 10^{2}$	$3.39 \times 10^{3}\backslash 3.20 \times 10^{2}$	$6.05 \times 10^{3}\backslash 5.46 \times 10^{2}$	$6.06 \times 10^{3}\backslash 4.89 \times 10^{2}$
F20	Mean\Std	$2.46 \times 10^{3}\backslash 8.32 \times 10^{1}$	$2.85 \times 10^{3}\backslash 6.82 \times 10^{1}$	$4.17 \times 10^{3}\backslash 1.82 \times 10^{2}$	$4.13 \times 10^{3}\backslash 1.86 \times 10^{2}$
F21	Mean\Std	$1.03 \times 10^{4}\backslash 1.09 \times 10^{3}$	$1.14 \times 10^{4}\backslash 1.04 \times 10^{3}$	$2.56 \times 10^{4}\backslash 1.24 \times 10^{3}$	$2.56 \times 10^{4}\backslash 1.48 \times 10^{3}$
F22	Mean\Std	$3.74 \times 10^{3}\backslash 2.31 \times 10^{2}$	$3.74 \times 10^{3}\backslash 1.77 \times 10^{2}$	$5.35 \times 10^{3}\backslash 3.78 \times 10^{2}$	$5.35 \times 10^{3}\backslash 2.91 \times 10^{2}$
F23	Mean\Std	$4.20 \times 10^{3}\backslash 2.15 \times 10^{2}$	$4.18 \times 10^{3}\backslash 1.76 \times 10^{2}$	$7.25 \times 10^{3}\backslash 5.84 \times 10^{2}$	$7.25 \times 10^{3}\backslash 4.62 \times 10^{2}$
F24	Mean\Std	$3.28 \times 10^{3}\backslash 6.13 \times 10^{1}$	$3.30 \times 10^{3}\backslash 7.30 \times 10^{1}$	$4.73 \times 10^{3}\backslash 2.76 \times 10^{2}$	$4.73 \times 10^{3}\backslash 2.60 \times 10^{2}$
F25	Mean\Std	$1.08 \times 10^{4}\backslash 1.79 \times 10^{3}$	$1.10 \times 10^{4}\backslash 1.63 \times 10^{3}$	$2.85 \times 10^{4}\backslash 2.46 \times 10^{3}$	$2.86 \times 10^{4}\backslash 4.26 \times 10^{3}$
F26	Mean\Std	$4.44 \times 10^{3}\backslash 5.43 \times 10^{2}$	$4.44 \times 10^{3}\backslash 3.71 \times 10^{2}$	$5.70 \times 10^{3}\backslash 1.25 \times 10^{3}$	$5.43 \times 10^{3}\backslash 6.12 \times 10^{2}$
F27	Mean\Std	$3.82 \times 10^{3}\backslash 1.43 \times 10^{2}$	$3.89 \times 10^{3}\backslash 1.77 \times 10^{2}$	$5.79 \times 10^{3}\backslash 4.35 \times 10^{2}$	$5.86 \times 10^{3}\backslash 4.88 \times 10^{2}$
F28	Mean\Std	$6.09 \times 10^{3}\backslash 7.55 \times 10^{2}$	$6.38 \times 10^{3}\backslash 7.33 \times 10^{2}$	$1.10 \times 10^{4}\backslash 1.01 \times 10^{3}$	$1.11 \times 10^{4}\backslash 9.60 \times 10^{2}$
F29	Mean\Std	$5.23 \times 10^{7}\backslash 1.82 \times 10^{7}$	$5.25 \times 10^{7}\backslash 1.55 \times 10^{7}$	$1.52 \times 10^{8}\backslash 6.75 \times 10^{7}$	$1.54 \times 10^{8}\backslash 7.05 \times 10^{7}$

5. Engineering Application

In this chapter, the proposed CSHHO is applied to model the reactive power output of a synchronous condenser. Due to the large number of UHV DC transmission projects in the power system, to ensure DC to DC power consumption and peaking demand, the scale of conventional units on the receiving AC grid is reduced, the dynamic reactive power support capability of the system is weakened, and the voltage stability margin is reduced [81]. This requires dynamic reactive power compensation devices to have instantaneous reactive power support characteristics in case of system failure, and the synchronous condenser to have reactive power output characteristics to meet the dynamic reactive power compensation requirements of the grid [82]. Modeling the reactive power support capability of a synchronous condenser is of great theoretical significance and practical value for the reactive power control of converter stations in high-voltage DC transmission systems with synchronous condenser.

The existing research methods for modeling the reactive power output of synchronous condenser are the mathematical analytical model calculation method and the experimental result fitting method [83–86] and both require large computational effort and have low accuracy, but few papers have studied the application of LSSVM in modeling the reactive power output of synchronous condenser. The advantages of the least squares support vector machine (LSSVM) are that it is less likely to fall into local minima and has high generalization ability [87]. Researchers have used various intelligent optimization algorithms to find the optimal results of kernel function parameters and regularization parameters, including the GA [88], Particle Swarm Optimization Algorithm (PSO) [89], Free Search Algorithm (FS) [90], Ant Colony Optimization Algorithm (ACO) [91], ABC Algorithm [92], GWO algorithm [93] and Backtracking Search Optimization Algorithm (BSA) [94], etc. However, the traditional swarm optimization algorithm is prone to the defects of falling into local optimum and low convergence accuracy in the search process. According to the results above, the CSHHO not only reduces the probability of the algorithm falling into local optimum and improves the convergence accuracy of the algorithm, but also has the advantages of the basic Harris Hawk optimization algorithm: 1. the steadiness of the searching cores; 2. the fruitfulness in the initial iterations; 3. the progressive selection scheme [5].

This paper proposes a CSHHO-LSSVM-based reactive power modeling method based on the numerical characteristics and global search capability of CSHHO. The optimal values of the penalty parameters, kernel function parameters, and loss function parameters of the LSSVM are found by using CSHHO to build the CSHHO-LSSVM model for the reactive power output of the synchronous condenser.

5.1. Principle of LSSVM

Support Vector Machine (SVM) is an ML method based on statistical learning theory, with kernel function as the core, which implicitly maps the data in the original space to the high-dimensional feature space, then finds the linear relationship in the feature space [87].

LSSVM is a regression algorithm that extends the basic SVM. Compared with the SVM algorithm, LSSVM requires fewer parameters and is more stable. LSSVM simplifies the complex constraints, which makes the improved SVM more capable of handling data. Moreover, by setting the error sum of squares as the loss function of the algorithm, LSSVM enhances the performance of regression prediction and improves the prediction accuracy. Simultaneously, the complexity of the algorithm is reduced, which reduces the processing time of the algorithm and provides more flexibility. LSSVM uses a nonlinear model on basis of SVM:

$$f(x) = (\omega, \phi(x)) + b \tag{28}$$

The input data were $(x_i, y_i)i = 1, \cdots l$ where $x_i \in R^d$ denoted the different elements, d denoted the dimension, $y_i \in R$ the expected value of the output, and l the total number of inputs. $\phi(x)$ denoted the mapping function. In summary, the LSSVM optimization objective was:

$$\min\frac{1}{2}\|\omega\|^2+\frac{1}{2}\gamma\sum_{i=1}^{l}e_i^2 \tag{29}$$

$s.t.\omega^T\varphi(x_i)+b+e_i=y_i(i=1,\cdots,l)$.

where e_i denoted the error, the magnitude of which determined the prediction accuracy;$e\in R^{l\times 1}$ denoted the error vector, γ denoted the regularization parameter r, which determined the magnitude of the error. Adding a Lagrangian multiplier to Equation (29),$\lambda\in R^{l\times 1}$, Equation (30) was expressed as:

$$\min J=\frac{1}{2}\|\omega\|^2+\frac{1}{2}\gamma\sum_{i=1}^{l}e_i^2-\sum_{i=1}^{l}\lambda_i\left(\omega^T\phi(x_i)+b+e_i-y_i\right) \tag{30}$$

From the KKT condition, we obtained:

$$\begin{cases}\frac{\partial J}{\partial\omega}=0\rightarrow\sum\limits_{i=1}^{l}\lambda_i\varphi(x_i)\\ \frac{\partial J}{\partial b}=0\rightarrow\sum\limits_{i=1}^{l}\lambda_i=0\\ \frac{\partial J}{\partial e_i}=0\rightarrow\lambda_i=\gamma e_i,i=1,2,\cdots,l\\ \frac{\partial J}{\partial\lambda_i}=0\rightarrow\omega^T\varphi(x_i)+b+e_i-y_i=0,i=1,2,\cdots,l\end{cases} \tag{31}$$

By eliminating the slack variables e_i and weight vectors ω, the optimization problem was linearized:

$$\begin{bmatrix}0 & \mathrm{Q}^{\mathrm{T}}\\ \mathrm{Q} & K+\frac{1}{\gamma}\mathrm{I}\end{bmatrix}\begin{bmatrix}b\\ \mathrm{A}\end{bmatrix}=\begin{bmatrix}0\\ \mathrm{Y}\end{bmatrix} \tag{32}$$

where $\mathrm{A}=[\alpha_1,\alpha_2,\cdots,\alpha_N]^{\mathrm{T}}$, $\mathrm{Q}=[1,1,\cdots,1]^{\mathrm{T}}$, was an $\mathrm{l}\times 1$ dimensional column vector, $\mathrm{Y}=[y_1,y_2,\cdots,y_N]^{\mathrm{T}}$. According to the Mercer condition, K denoted a kernel function: $K(x_i,x_j)=\boldsymbol{\varphi}(x_i)^{\mathrm{T}}\boldsymbol{\varphi}(x_j)\ i,j=1,2,\cdots,N$. The Radial Basis Function kernel function was chosen for the model:

$$k(x_i,x_j)=\exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma^2}\right)\ \sigma>0 \tag{33}$$

Therefore, the nonlinear prediction model was expressed by Equation (34):

$$y=\sum_{i=1}^{l}\lambda_i K(x_i,x)+b \tag{34}$$

When predicting with least squares support vector regression models, the penalty factor and radial basis kernel function parameters were the two parameters to be solved.

5.2. Simulation and Verification

The reactive power regulation results of a synchronous condenser based on PSCAD/EMTDC simulation software were used as training samples and test samples. Table 11 shows the main parameters. The data with serial numbers 9, 14, 26 and 35 in Table 11 were taken as the test samples, and the rest were the training samples.

First, the data were preprocessed, and the LSSVM was trained by using CSHHO to find the penalty parameters, kernel function parameters and optimal parameters of the loss function (γ, σ, S), and the LSSVM was predicted by applying the test sample to the LSSVM, and the regression fitted prediction model was output. The algorithm flow is shown in Figure 6. Figure 7 compares the output results of the LSSVM model of the test sample and the CSHHO-LSSVM model were compared with those of the test sample and the errors of the LSSVM model of the test sample and the CSHHO-LSSVM model. The output regression of CSHHO-LSSVM model was better fitted and had higher accuracy and radial basis kernel function parameters.

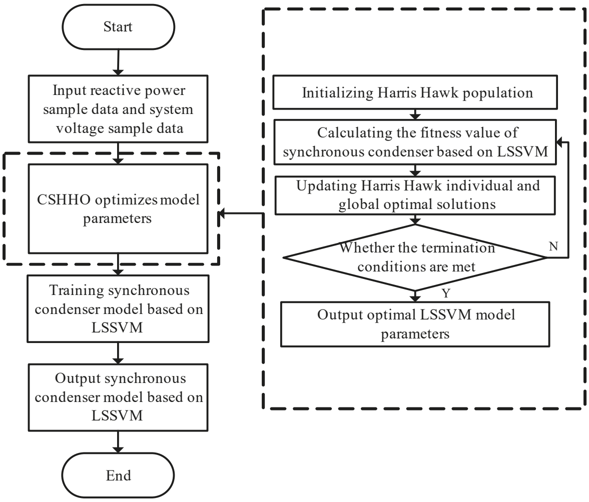

Start
Input reactive power sample data and system voltage sample data
CSHHO optimizes model parameters
Training synchronous condenser model based on LSSVM
Output synchronous condenser model based on LSSVM
End
Initializing Harris Hawk population
Calculating the fitness value of synchronous condenser based on LSSVM
Updating Harris Hawk individual and global optimal solutions
Whether the termination conditions are met
N
Y
Output optimal LSSVM model parameters

Figure 6. CSHHO Optimizes Synchronous condenser Reactive Power Output based LSSVM.

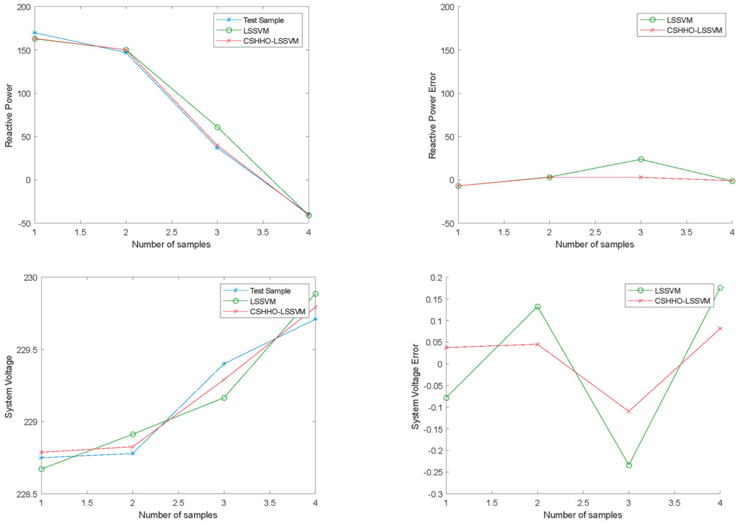


Figure 7. Output and error comparison diagram of LSSVM model and CSHHO-LSSVM model.

To verify the generalization ability of the CSHHO-LSSVM model, it was evaluated by absolute deviation, as shown in Table 13. The table shows the absolute deviation range of CSHHO-LSSVM model from 0.0123 to 0.989, indicating that the accuracy of the CSHHO-LSSVM model was high.

Table 13. Training samples and test samples.

Input Sample		Output Sample	
Excitation Current/A	**Exciting Voltage/V**	**Reactive Power/Mvar**	**System Voltage/kV**
25.488	29.267	305.0	228.4
23.364	26.103	261.5	228.2
21.594	24.521	232.5	228.5
20.154	22.939	212.5	228.5
19.6824	22.148	197.5	228.6
18.8328	21.159	186.4	228.6
18.408	20.408	177.4	228.7
17.9124	19.9728	170.2	228.75
17.7	19.775	163.8	228.8
17.346	19.3795	158.5	228.8
16.992	18.984	154.2	228.8
16.7088	18.7863	150.5	228.8
16.638	18.5727	147.0	228.78
16.461	18.3987	144.2	228.85
16.1424	17.9715	137.4	228.84
15.6822	17.402	131.1	228.9
15.222	17.0065	125.0	228.9
14.868	16.611	118.35	229
14.2308	15.9782	108.75	229
12.9564	14.5346	88.5	229.1
12.39	13.8425	78.3	229.1
11.682	12.953	67.8	229.2
10.974	12.458	57.5	229.3
10.3368	11.4695	47.5	229.3
9.912	11.074	37.0	229.4
8.9916	10.283	27.5	229.4
8.4252	9.1756	17.2	229.48
7.8588	8.8592	7.5	229.5
7.2924	8.3055	1.75	229.57
6.9738	7.91	−2.5	229.6
6.372	6.9213	−11.7	229.6
5.8056	6.7235	−21.0	229.62
5.664	6.4467	−30.5	229.7
4.956	5.1415	−39.6	229.71
4.248	4.5483	−48.5	229.8
3.54	4.351	−57.5	229.875
2.832	3.164	−57.5	229.875
2.124	2.4521	−66.5	229.9

From the reactive power and system voltage simulation results of the synchronous condenser in Tables 14–16, we can see that the maximum absolute error of the reactive power simulation result of CSHHO-LSSVM model was 0. 989 Mvar, and the maximum absolute error of the system voltage simulation result was 0.0415 kV, which were smaller than the simulation results of LSSVM model, indicating that the CSHHO-LSSVM model had higher accuracy and better regression fitting performance. The LSSVM model was more accurate and had better regression fitting performance.

Table 14. CSHHO-LSSV Model generalization ability verification.

		Prediction Value	Sample Value	Absolute Error	Relative Error
9	Reactive Power/MVar	163.0515	163.8	−0.7485	−0.4570
	System Voltage/kV	228.7877	228.8	−0.0123	−0.0054
14	Reactive Power/MVar	144.5989	144.2	0.3989	0.2766
	System Voltage/kV	228.8882	228.85	0.0382	0.0167
26	Reactive Power/MVar	28.1322	27.6	0.5322	1.9282
	System Voltage/kV	229.4014	229.45	−0.0486	−0.0212
35	Reactive Power/MVar	−40.5890	−39.6	−0.989	2.4975
	System Voltage/kV	229.7915	229.75	0.0415	0.0181

Table 15. Reactive power simulation results comparison.

		Prediction Value	Sample Value	Absolute Error	Relative Error
LSSVM	9	162.9631	163.8	−0.8369	−0.5128
	35	−40.8531	−39.6	−1.2531	3.1649
CSHHO−LSSVM	9	163.0515	163.8	−0.7485	−0.4570
	35	−40.5890	−39.6	−0.989	2.4975

Table 16. System voltage simulation results comparison.

		Prediction Value	Sample Value	Absolute Error	Relative Error
LSSVM	9	228.6717	228.8	−0.1283	−0.0561
	35	229.8859	229.75	0.1359	0.0591
CSHHO−LSSVM	9	228.7877	228.8	−0.0123	−0.0054
	35	229.7915	229.75	0.0415	0.0181

6. Conclusions

Here, we analyzed the shortcomings of the basic HHO algorithm and applied the chaotic mapping population initialization, adaptive weighting, variable spiral position update and optimal neighborhood disturbance mechanisms to the classical HHO algorithm, in which the Gauss chaotic mapping population initialization increased the coverage of the solution space by the initial solution of the algorithm, the adaptive weighting mechanism sped up the movement of Harris hawk populations to the optimal solution, and the variable spiral position update increased the ability of Harris hawk populations. The optimal neighborhood disturbance mechanism helped the improved algorithm to increase the algorithm's global search capability and avoided premature maturity. To verify the optimal performance of the four strategies, the experiments were separated into two groups.

First, seven commonly used chaotic mappings were selected for the population initialization of the HHO algorithm, including Sinusoidal, Tent, Kent, Cubic, Logistic, Gauss, and Circle mappings. The HHO algorithm's performance after population initialization of each of these seven mappings was evaluated. The HHO algorithm's performance after population initialization of Gauss mapping was significantly better than that of the HHO algorithm after population initialization of other mappings in terms of solution accuracy. Second, based on the results of the first set of experiments, the Gauss mapping was used for population initialization, and adaptive weights, variable spiral position update, and optimal neighborhood disturbance mechanisms were introduced into the algorithm after population initialization. Next, CSHHO was compared with other classical algorithms including WOA, SCA, CSO and advanced algorithms including GCHHO and DEPSOASS and Improved GSA and DGOBLFOA based on 23 classical test functions and the means and standard deviations of all algorithms were analyzed. Subsequently, each algorithm's performance was evaluated comprehensively using Friedman’s test and the Bonferroni–Holm corrected Wilcoxon signed-rank test with 5% confidence level, where numerical analysis concluded that CSHHO outperformed the other algorithms. In detail, analyzing the experimental results of this work, in the population initialization phase, Gauss chaos mapping had the best results in F2, F6, F12, F17, and F23 test functions, and comparing the results of the remaining six chaotic mappings, Gauss chaos mapping obtained the most optimal solutions; CSHHO algorithm outperformed HHO in 17 benchmark functions out of 23 classical test functions, outperformed WOA in 21 results, SCA in 23 results, and CSO in 22 experiments. It outperformed GCHHO in 9 results. Meanwhile, in the statistical experiments of advanced meta-heuristic and classical meta-heuristic, the ARVs obtained by CSHHO were 1.93 and 2.57, respectively, which were lower than the values obtained by other pairwise meta-heuristics in the same group of experiments. Additionally, dimensional scalability tests were conducted for CSHHO on the IEEECEC2017 dataset, including 50 and 100 dimensions, and the results showed that the improved optimizer effectively handled high-dimensional data with good stability. Meanwhile, in the statistical experiments of advanced meta-heuristic and classical meta-heuristic, CSHHO obtained ARVs of 1.70 and 2.57, respectively, which were lower than the values obtained by other meta-heuristic algorithms in the same set of experiments. Furthermore, dimensional scalability tests were conducted for CSHHO on the IEEE CEC 2017 dataset, including 50 and 100 dimensions, and the results showed that the improved optimizer effectively handled high-dimensional data with excellent stability.

Here, the CSHHO algorithm was also applied to the engineering problem of reactive power output modeling of the synchronous condenser. In view of the defects of the many calculations and low accuracy of the traditional reactive power output modeling method of the synchronous condenser, CSHHO-LSSVM was used to model the reactive power output of the synchronous condenser based on the advantages of LSSVM, which was not easy to fall into local minimum and had strong generalization ability, and CSHHO had high search accuracy and strong global search ability. The excitation current and excitation voltage of the synchronous condenser were used as the input of the LSSVM model, and the reactive power and system voltage were used as the LSSVM model’s output. CSHHO was used to find the optimal values of the penalty parameter, kernel function parameter, and loss function parameter of LSSVM. The experiment showed that the CSHHHO-LSSVM model had better accuracy and better regression fitting performance compared with LSSVM.

In future work, we will try to improve the convergence speed and search accuracy of the algorithm and balance the exploration and exploitation phases of the algorithm to obtain better search performance. Additionally, the next step will be to investigate how CSHHO can be used to solve multi-objective optimization problems. In addition, CSHHO can also be used for evolutionary ML, such as extreme learning machines and parameter tuning of convolution neural networks. Other problems include grid scheduling and 3D multi-objective tracking.

Author Contributions: Methodology, C.W.; resources, S.J.; funding acquisition, S.J., R.G., Y.L. and Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (No. 61871318), the Shaanxi Provincial Key Research and Development Project (No. 2019GY-099) and Open project of Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing (No. 2020CP10).

Institutional Review Board Statement: The study did not involve human and animals.

Informed Consent Statement: The study did not involve human and animals.

Data Availability Statement: In the paper, all the data generation information has been given in detail in the related chapter.

Acknowledgments: The author thanks the referees for detailed and constructive criticism of the original manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. The classical 23 test functions.

Function	Dimensions	Range	f_{min}
unimodal benchmark functions			
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30,100, 500, 1000	[100, 100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30,100, 500, 1000	[10, 10]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30,100, 500, 1000	[100, 100]	0
$f_4(x) = \max_i \{ \lvert x_i \rvert, 1 \le i \le n \}$	30,100, 500, 1000	[100, 100]	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	30,100, 500, 1000	[30, 30]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30,100, 500, 1000	[100, 100]	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	30,100, 500, 1000	[128, 128]	0
multimodal benchmark functions			
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert} \right)$	30,100, 500, 1000	[500, 500]	−418.9829 × n
$f_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10 \cos(2\pi x_i) + 10 \right]$	30,100, 500, 1000	[5.12, 5.12]	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right)$ $- \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	30,100, 500, 1000	[32, 32]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	30,100, 500, 1000	[600, 600]	0
$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \right.$ $\left. \left[1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30,100, 500, 1000	[50, 50]	0
$y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a & < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1) \right] \right.$ $\left. + (x_n - 1)^2 \left[1 + \sin^2(2\pi x_n) \right] \right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30,100, 500, 1000	[50, 50]	0
fixed-dimension multimodal benchmark functions			
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} \left(x_i - a_{ij} \right)^6} \right)^{-1}$	2	[−65, 65]	1
$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 \left(b_i^2 + b_i x_2 \right)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[−5, 5]	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3} x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	[−5, 5]	0.398
$f_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2 \right) \right]$ $\times \left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2 \right) \right]$	2	[−2, 2]	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij} \left(x_j - p_{ij} \right)^2 \right)$	3	[1, 3]	−3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij} \left(x_j - p_{ij} \right)^2 \right)$	6	[0, 1]	−3.32
$f_{21}(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	−10.1532
$f_{22}(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	−10.4028
$f_{23}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	−10.5363

References

1. Kundu, T.; Garg, H. A Hybrid ITLHHO Algorithm for Numerical and Engineering Optimization Problems. *Int. J. Intell. Syst.* **2021**, *36*, 1–81. [CrossRef]
2. Garg, H. A Hybrid GSA-GA Algorithm for Constrained Optimization Problems. *Inf. Sci.* **2019**, *478*, 499–523. [CrossRef]
3. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [CrossRef]
4. Wu, Y. A Survey on Population-Based Meta-Heuristic Algorithms for Motion Planning of Aircraft. *Swarm Evol. Comput.* **2021**, *62*, 100844. [CrossRef]
5. Alabool, H.M.; Alarabiat, D.; Abualigah, L.; Heidari, A.A. Harris hawks optimization: A comprehensive review of recent variants and applications. *Neural Comput. Appl.* **2021**, *33*, 8939–8980. [CrossRef]
6. Vasant, P. *Handbook of Research on Artificial Intelligence Techniques and Algorithms*; IGI Global: Hershey, PA, USA, 2015. [CrossRef]
7. Simon, D. *Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence*; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2013; ISBN 978-0-470-93741-9.
8. Dréo, J. (Ed.) *Metaheuristics for Hard Optimization: Methods and Case Studies*; Springer: Berlin, Germany, 2006; ISBN 978-3-540-23022-9.
9. Salcedo-Sanz, S. Modern Meta-Heuristics Based on Nonlinear Physics Processes: A Review of Models and Design Procedures. *Phys. Rep.* **2016**, *655*, 1–70. [CrossRef]
10. Adeyanju, O.M.; Canha, L.N. Decentralized Multi-Area Multi-Agent Economic Dispatch Model Using Select Meta-Heuristic Optimization Algorithms. *Electr. Power Syst. Res.* **2021**, *195*, 107128. [CrossRef]
11. Zhu, M.; Chu, S.-C.; Yang, Q.; Li, W.; Pan, J.-S. Compact Sine Cosine Algorithm with Multigroup and Multistrategy for Dispatching System of Public Transit Vehicles. *J. Adv. Transp.* **2021**, *2021*, 1–16. [CrossRef]
12. Fu, X.; Fortino, G.; Li, W.; Pace, P.; Yang, Y. WSNs-Assisted Opportunistic Network for Low-Latency Message Forwarding in Sparse Settings. *Future Gener. Comput. Syst.* **2019**, *91*, 223–237. [CrossRef]
13. Dhiman, G. SSC: A Hybrid Nature-Inspired Meta-Heuristic Optimization Algorithm for Engineering Applications. *Knowl.-Based Syst.* **2021**, *222*, 106926. [CrossRef]
14. Han, Y.; Gu, X. Improved Multipopulation Discrete Differential Evolution Algorithm for the Scheduling of Multipurpose Batch Plants. *Ind. Eng. Chem. Res.* **2021**, *60*, 5530–5547. [CrossRef]
15. Loukil, T.; Teghem, J.; Tuyttens, D. Solving multi-objective production scheduling problems using meta-heuristics. *Eur. J.Oper. Res.* **2005**, *161*, 42–61. [CrossRef]
16. Li, Q.; Chen, H.; Huang, H.; Zhao, X.; Cai, Z.; Tong, C.; Liu, W.; Tian, X. An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis. *Comput. Math. Methods Med.* **2017**, 1–15. [CrossRef]
17. Li, Y.; Liu, J.; Tang, Z.; Lei, B. Deep Spatial-Temporal Feature Fusion From Adaptive Dynamic Functional Connectivity for MCI Identification. *IEEE Trans. Med. Imaging.* **2020**, *39*, 2818–2830. [CrossRef] [PubMed]
18. Corazza, M.; di Tollo, G.; Fasano, G.; Pesenti, R. A Novel Hybrid PSO-Based Metaheuristic for Costly Portfolio Selection Problems. *Ann. Oper. Res.* **2021**, *304*, 109–137. [CrossRef]
19. Gaspero, L.D.; Tollo, G.D.; Roli, A.; Schaerf, A. Hybrid Metaheuristics for Constrained Portfolio Selection Problems. *Quant. Financ.* **2011**, *11*, 1473–1487. [CrossRef]
20. Shen, L.; Chen, H.; Yu, Z.; Kang, W.; Zhang, B.; Li, H.; Yang, B.; Liu, D. Evolving Support Vector Machines Using Fruit Fly Optimization for Medical Data Classification. *Knowl.-Based Syst.* **2016**, *96*, 61–75. [CrossRef]
21. Wang, M.; Chen, H.; Yang, B.; Zhao, X.; Hu, L.; Cai, Z.; Huang, H.; Tong, C. Toward an Optimal Kernel Extreme Learning Machine Using a Chaotic Moth-Flame Optimization Strategy with Applications in Medical Diagnoses. *Neurocomputing* **2017**, *267*, 69–84. [CrossRef]
22. Song, J.; Zheng, W.X.; Niu, Y. Self-Triggered Sliding Mode Control for Networked PMSM Speed Regulation System: A PSO-Optimized Super-Twisting Algorithm. *IEEE Trans. Ind. Electron.* **2021**, *69*, 763–773. [CrossRef]
23. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
24. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gen. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
25. Dong, W.; Zhou, M. A Supervised Learning and Control Method to Improve Particle Swarm Optimization Algorithms. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1135–1148. [CrossRef]
26. Mareda, T.; Gaudard, L.; Romerio, F. A Parametric Genetic Algorithm Approach to Assess Complementary Options of Large Scale Windsolar Coupling. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 260–272. [CrossRef]
27. Jian, Z.; Liu, S.; Zhou, M. Modified cuckoo search algorithm to solve economic power dispatch optimization problems. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 794–806.
28. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science.* **1983**, *220*, 671–680. [CrossRef]
29. Xing, B.; Gao, W.-J. Gravitational Search Algorithm. In *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*; Intelligent Systems Reference Library; Springer International Publishing: Cham, Germany, 2014; Volume 62, pp. 355–364. ISBN 978-3-319-03403-4.
30. Alatas, B. ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Syst. Appl.* **2011**, *38*, 13170–13180. [CrossRef]
31. Patel, V.K.; Savsani, V.J. Heat transfer search (HTS): A novel optimization algorithm. *Inf. Sci.* **2015**, *324*, 217–246. [CrossRef]

32. Abdechiri, M.; Meybodi, M.R.; Bahrami, H. Gases brownian motion optimization: An algorithm for optimization (GBMO). *Appl. Soft Comput.* **2013**, *13*, 2932–2946. [CrossRef]
33. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener. Comput. Syst.* **2019**, *101*, 646–667. [CrossRef]
34. Holland, J.H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66–72. [CrossRef]
35. Schneider, B.; Ranft, U. *Simulationsmethoden in der Medizin und Biologie*; Springer: Berlin/Heidelberg, Germany, 1978.
36. Yang, X.-S. Differential evolution. In *Nature-Inspired Optimization Algorithms. Algorithms*; Elsevier: London UK, 2021; Volume 6, pp. 101–109.
37. Storn, R.; Price, K. Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
38. Simon, D. Biogeography-Based Optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [CrossRef]
39. Karaboga, D.; Akay, B. A comparative study of Artificial Bee Colony algorithm. *Appl. Math. Comput.* **2009**, *214*, 108–132. [CrossRef]
40. Yang, X.S. Firefly algorithm, stochastic test functions and design optimization. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78. [CrossRef]
41. Jiang, X.; Li, S. BAS: Beetle antennae search algorithm for optimization problems. *Int. J. Robot. Control* **2018**, *1*, 1. [CrossRef]
42. Okwu, M.O. (Ed.) *Grey Wolf Optimizer, Metaheuristic Optimization, Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications*; Springer International Publishing: Cham, Switzerland, 2021. [CrossRef]
43. Li, M.D.; Zhao, H.; Weng, X.W.; Han, T. A novel nature-inspired algorithm for optimization: Virus colony search. *Adv. Eng. Softw.* **2016**, *92*, 65–88. [CrossRef]
44. Glover, F. Tabu search—Part I. *ORSA J. Comput.* **1989**, *1*, 190–206. [CrossRef]
45. Kumar, M.; Kulkarni, A.J.; Satapathy, S.C. Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology. *Future Gener. Comput. Syst.* **2018**, *81*, 252–272. [CrossRef]
46. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching–learning-based optimization: An optimization method for continuous non-linear large scale problems. *Inf. Sci.* **2012**, *183*, 1–15. [CrossRef]
47. Hosseini, S.; Al Khaled, A. A Survey on the Imperialist Competitive Algorithm Metaheuristic: Implementation in Engineering Domain and Directions for Future Research. *Appl. Soft. Comput.* **2014**, *24*, 1078–1094. [CrossRef]
48. Jia, H.; Peng, X.; Kang, L.; Li, Y.; Jiang, Z.; Sun, K. Pulse coupled neural network based on Harris hawks optimization algorithm for image segmentation. *Multimed Tools Appl.* **2020**, *79*, 28369–28392. [CrossRef]
49. Fan, C.; Zhou, Y.; Tang, Z. Neighborhood centroid opposite-based learning harris hawks optimization for training neural networks. *Evol. Intell.* **2020**, *14*, 1847–1867. [CrossRef]
50. Saravanan, G.; Ibrahim, A.M.; Kumar, D.S.; Vanitha, U.; Chandrika, V.S. Iot Based Speed Control of BLDC Motor with Harris Hawks Optimization Controller. *Int. J. Grid Distrib. Comput.* **2020**, *13*, 1902–1919.
51. Qu, C.; He, W.; Peng, X.; Peng, X. Harris hawks optimization with information exchange. *Appl. Math. Model.* **2020**, *84*, 52–75. [CrossRef]
52. Devarapalli, R.; Bhattacharyya, B. Application of modified harris hawks Optimization in power system oscillations damping controller design. In Proceedings of the 2019 8th International Conference on Power Systems (ICPS), Jaipur, India, 20–22 December 2019.
53. Elgamal, Z.M.; Yasin, N.B.M.; Tubishat, M.; Alswaitti, M.; Mirjalili, S. An improved harris hawks optimization algorithm with simulated annealing for feature selection in the Medical Field. *IEEE Access.* **2020**, *8*, 186638–186652. [CrossRef]
54. Song, S.; Wang, P.; Heidari, A.A.; Wang, M.; Zhao, X.; Chen, H.; He, W.; Xu, S. Dimension decided harris hawks optimization with gaussian mutation: Balance analysis and diversity patterns. *Knowl.-Based Syst.* **2021**, *215*, 106425. [CrossRef]
55. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–83. [CrossRef]
56. Bednarz, J.C. Cooperative Hunting Harris' Hawks (Parabuteo unicinctus). *Science* **1988**, *239*, 1525–1527. [CrossRef] [PubMed]
57. Lefebvre, L.; Whittle, P.; Lascaris, E.; Finkelstein, A. Feeding innovations and forebrain size in birds. *Anim. Behav.* **1997**, *53*, 549–560. [CrossRef]
58. Sol, D.; Duncan, R.P.; Blackburn, T.M.; Cassey, P.; Lefebvre, L. Big brains, Enhanced Cognition, and Response of birds to Novel environments. *Proc. Natl. Acad. Sci. USA* **2005**, *102*, 5460–5465. [CrossRef] [PubMed]
59. Kazimipour, B.; Li, X.; Qin, A.K. A review of population initialization techniques for evolutionary algorithms. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 2585–2592. [CrossRef]
60. Migallón, H.; Jimeno-Morenilla, A.; Rico, H.; Sánchez-Romero, J.L.; Belazi, A. Multi-Level Parallel Chaotic Jaya Optimization Algorithms for Solving Constrained Engineering Design Problems. *J. Supercomput.* **2021**, *77*, 12280–12319. [CrossRef]
61. Alatas, B.; Akin, E. Multi-Objective Rule Mining Using a Chaotic Particle Swarm Optimization Algorithm. *Knowl.-Based Syst.* **2009**, *22*, 455–460. [CrossRef]
62. Arora, S.; Anand, P. Chaotic Grasshopper Optimization Algorithm for Global Optimization. *Neural Comput. Appl.* **2019**, *31*, 4385–4405. [CrossRef]
63. Wang, G.-G.; Guo, L.; Gandomi, A.H.; Hao, G.-S.; Wang, H. Chaotic Krill Herd Algorithm. *Inf. Sci.* **2014**, *274*, 17–34. [CrossRef]
64. Mitić, M.; Vuković, N.; Petrović, M.; Miljković, Z. Chaotic Fruit Fly Optimization Algorithm. *Knowl.-Based Syst.* **2015**, *89*, 446–458. [CrossRef]

65. Kumar, Y.; Singh, P.K. A Chaotic Teaching Learning Based Optimization Algorithm for Clustering Problems. *Appl. Intell.* **2019**, *49*, 1036–1062. [CrossRef]
66. Pierezan, J.; dos Santos Coelho, L.; Mariani, V.C.; de Vasconcelos Segundo, E.H.; Prayogo, D. Chaotic Coyote Algorithm Applied to Truss Optimization Problems. *Comput. Struct.* **2021**, *242*, 106353. [CrossRef]
67. Sayed, G.I.; Tharwat, A.; Hassanien, A.E. Chaotic Dragonfly Algorithm: An Improved Metaheuristic Algorithm for Feature Selection. *Appl. Intell.* **2019**, *49*, 188–205. [CrossRef]
68. Chen, K.; Zhou, F.; Liu, A. Chaotic Dynamic Weight Particle Swarm Optimization for Numerical Function Optimization. *Knowl.-Based Syst.* **2018**, *139*, 23–40. [CrossRef]
69. Anand, P.; Arora, S. A Novel Chaotic Selfish Herd Optimizer for Global Optimization and Feature Selection. *Artif. Intell. Rev.* **2020**, *53*, 1441–1486. [CrossRef]
70. Yao, X.; Liu, Y.; Lin, G. Evolutionary Programming Made Faster. *IEEE Trans. Evol. Comput.* **1999**, *3*, 82–102. [CrossRef]
71. Digalakis, J.G.; Margaritis, K.G. On benchmarking functions for genetic algorithms. *Int. J. Comput. Math.* **2001**, *77*, 481–506. [CrossRef]
72. Chen, H.; Zhang, Q.; Luo, J. An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine. *Appl. Soft. Comput.* **2020**, *86*, 105–884. [CrossRef]
73. Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]
74. Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A new bio-inspired algorithm: Chicken swarm optimization. In *Advances in Swarm Intelligence*; Tan, Y., Shi, Y., Coello, C.A.C., Eds.; Springer International Publishing: Cham, Germany, 2014; pp. 86–94. [CrossRef]
75. Zhang, J.; Chen, J.; Che, L. Hybrid PSO Algorithm with Adaptive Step Search in Noisy and Noise-Free Environments. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8.
76. Jordehi, A.R. Gravitational Search Algorithm with Linearly Decreasing Gravitational Constant for Parameter Estimation of Photovoltaic Cells. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), Donostia, Spain, 5–8 June 2017; pp. 37–42.
77. Feng, X.; Liu, A.; Sun, W.; Yue, X.; Liu, B. A Dynamic Generalized Opposition-Based Learning Fruit Fly Algorithm for Function Optimization. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–7.
78. Demsar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
79. García, S.; Fernández, A.; Luengo, J.; Herrera, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.* **2010**, *180*, 2044–2064. [CrossRef]
80. Groppe, D.M.; Urbach, T.P.; Kutas, M. Mass univariate analysis of event-related brain potentials/fields I: A critical tutorial review. *Psychophysiology* **2011**, *48*, 1711–1725. [CrossRef] [PubMed]
81. Deliparaschos, K.M.; Nenedakis, F.I.; Tzafestas, S.G. Design and implementation of a fast digital fuzzy logic controller using FPGA Technology. *J. Intell. Robot. Syst.* **2006**, *45*, 77–96. [CrossRef]
82. Sapkota, B.; Vittal, V. Dynamic VAr planning in a large power system using trajectory sensitivities. *IEEE Trans. Power Syst.* **2010**, *25*, 461–469. [CrossRef]
83. Huang, H.; Xu, Z.; Lin, X. Improving performance of Multi-infeed HVDC systems using grid dynamic segmentation technique based on fault current limiters. *IEEE Trans. Power Syst.* **2012**, *27*, 1664–1672. [CrossRef]
84. Yong, T. A discussion about standard parameter models of synchronous machine. *Power Syst. Technol.* **2007**, *12*, 47.
85. Grigsby, L.L. (Ed.) *Power system Stability and Control*, 3rd ed.; Taylor & Francis: Boca Raton, FL, USA, 2012.
86. Tian, Z. Backtracking search optimization algorithm-based least square support vector machine and its applications. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103801. [CrossRef]
87. Suykens, J.A.K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [CrossRef]
88. Zendehboudi, A. Implementation of GA-LSSVM modelling approach for estimating the performance of solid desiccant wheels. *Energy Convers. Manag.* **2016**, *127*, 245–255. [CrossRef]
89. Chamkalani, A.; Zendehboudi, S.; Bahadori, A.; Kharrat, R.; Chamkalani, R.; James, L.; Chatzis, I. Integration of LSSVM technique with PSO to determine asphaltene deposition. *J. Pet. Sci. Eng.* **2014**, *124*, 243–253. [CrossRef]
90. Zhongda, T.; Shujiang, L.; Yanhong, W.; Yi, S. A prediction method based on wavelet transform and multiple models fusion for chaotic time series. *Chaos Solitons Fractals* **2017**, *98*, 158–172. [CrossRef]
91. Liu, W.; Zhang, X. Research on the supply chain risk assessment based on the improved LSSVM algorithm. *Int. J. U-Serv. Sci. Technol.* **2016**, *9*, 297–306. [CrossRef]
92. Jain, S.; Bajaj, V.; Kumar, A. Efficient algorithm for classification of electrocardiogram beats based on artificial bee colony-based least-squares support vector machines classifier. *Electron. Lett.* **2016**, *52*, 1198–1200. [CrossRef]
93. Yang, A.; Li, W.; Yang, X. Short-term electricity load forecasting based on feature selection and Least Squares Support Vector Machines. *Knowl.-Based Syst.* **2019**, *163*, 159–173. [CrossRef]
94. Adankon, M.M.; Cheriet, M. Support vector machine. In *Encyclopedia of Biometrics*; Springer: Boston, MA, USA, 2014; Volume 3, pp. 1–9. [CrossRef]

symmetry

MDPI

Article

Formation of Fuzzy Patterns in Logical Analysis of Data Using a Multi-Criteria Genetic Algorithm

Igor S. Masich [1,2], Margarita A. Kulachenko [1], Predrag S. Stanimirović [3], Aleksey M. Popov [1], Elena M. Tovbis [1], Alena A. Stupina [1,4] and Lev A. Kazakovtsev [1,4,*]

[1] Institute of Informatics and Telecommunications, Reshetnev Siberian State University of Science and Technology, 31 Krasnoyarsky Rabochy av., 660037 Krasnoyarsk, Russia; i-masich@yandex.ru (I.S.M.); margaritakulachenko@yandex.ru (M.A.K.); vm_popov@sibsau.ru (A.M.P.); sibstu2006@rambler.ru (E.M.T.); h677hm@gmail.com (A.A.S.)

[2] Institute of Space and Information Technologies, Siberian Federal University, 79 Svobodny pr., 660041 Krasnoyarsk, Russia

[3] Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia; pecko@pmf.ni.ac.rs

[4] Institute of Business Process Management, Siberian Federal University, 79 Svobodny pr., 660041 Krasnoyarsk, Russia

* Correspondence: levk@bk.ru

Citation: Masich, I.S.; Kulachenko, M.A.; Stanimirović, P.S.; Popov, A.M.; Tovbis, E.M.; Stupina, A.A.; Kazakovtsev, L.A. Formation of Fuzzy Patterns in Logical Analysis of Data Using a Multi-Criteria Genetic Algorithm. *Symmetry* **2022**, *14*, 600. https://doi.org/10.3390/sym14030600

Academic Editor: László T. Kóczy

Received: 25 February 2022
Accepted: 15 March 2022
Published: 17 March 2022

Abstract: The formation of patterns is one of the main stages in logical data analysis. Fuzzy approaches to pattern generation in logical analysis of data allow the pattern to cover not only objects of the target class, but also a certain proportion of objects of the opposite class. In this case, pattern search is an optimization problem with the maximum coverage of the target class as an objective function, and some allowed coverage of the opposite class as a constraint. We propose a more flexible and symmetric optimization model which does not impose a strict restriction on the pattern coverage of the opposite class observations. Instead, our model converts such a restriction (purity restriction) into an additional criterion. Both, coverage of the target class and the opposite class are two objective functions of the optimization problem. The search for a balance of these criteria is the essence of the proposed optimization method. We propose a modified evolutionary algorithm based on the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) to solve this problem. The new algorithm uses pattern formation as an approximation of the Pareto set and considers the solution's representation in logical analysis of data and the informativeness of patterns. We have tested our approach on two applied medical problems of classification under conditions of sample asymmetry: one class significantly dominated the other. The classification results were comparable and, in some cases, better than the results of commonly used machine learning algorithms in terms of accuracy, without losing the interpretability.

Keywords: logical analysis of data; pattern generation; genetic algorithm

1. Introduction

Logical analysis of data (LAD) is a methodology for processing a set of observations, or objects, some of which belong to a specific subset (positive observations), and the rest does not belong to it (negative observations) [1]. These observations are described by features, generally numerical, nominal, or binary. Logical analysis of data is performed by detecting pattern-logical expressions that are true for positive (or negative) observations and not performed for negative (or, respectively, positive) observations [2]. Thus, regions in the feature space containing observations of the corresponding classes can be approximated using a set of positive and negative patterns. To identify such patterns, we use models and methods of combinatorial optimization [3–5].

Classification problems are one of the application fields of LAD [6,7]. From the point of view of solving classification problems, applying LAD can be considered as the construction of a rule-based classifier. Like other rule-based classification approaches, such as decision trees and lists of rules, this approach has the advantage that it constructs a "transparent" classifier. Thus, it belongs to interpretive machine learning methods.

Two types of patterns can be distinguished [8]. The first type is homogeneous (pure, clear) patterns. A homogeneous pattern covers part of the observations of a particular class (for example, positive) and does not cover any observation of another class (negative). However, if we consider real data for constructing a classifier, then pure patterns often do not give a good result. Due to the noisiness of the data, the presence of inaccuracies, errors, and outliers, pure patterns may have too low generalizing ability, and their use leads to overfitting. In such cases, the best results are shown by fuzzy (partial) patterns, in which the homogeneity constraint is weakened. Such weakening (relaxation) leads to the formation of more generalized patterns [4,9]. The pattern search problem is considered as an optimization problem, where the objective function is the number of covered observations of a certain class under a relaxed non-coverage constraint of observations of the opposite class.

Modern literature offers various approaches to the formation of patterns [8]. To generate patterns, enumeration-based algorithms [6,10], algorithms based on integer programming [2] or mixed approach based on both integer and linear programming principles [11,12] are used. In [5], a genetic algorithm for generating patterns is described. In [13], an approach based on metaheuristics is presented, in which the key idea is the construction of a pool of patterns for each given observation of the training set. Logical analysis of data is used in many application areas, such as cancer diagnosis and coronary risk prediction [2,10,11,14], credit risk rating [11,15–17], assessment of the potential of an economy to attract foreign direct investment [18], predicting the number of airline passengers [19], fault prognosis and anomaly detection [20–23], and others.

Thus, in traditional approaches to the logical analysis of data with homogeneous patterns, each pattern covers part of the observations of the target class and no observations of the opposite class. Otherwise, the homogeneity constraint is transformed into a relaxed non-coverage constraint related to a number or ratio of observations of the opposite class. Thus, such optimization model is not symmetric in the sense that the problem is focused on the number of covered observations of the target class while the coverage of the opposite class is considered as a constraint set at a certain level. Such an approach with the fuzzy patterns remains in the domain of the single-criterion optimization.

Recently, fuzzy logic theory has been widely developed in research. As mentioned in the literature [13], a certain degree of fuzziness seems to improve the robustness of the classification algorithm. In a fuzzy classification system, an object can be classified by applying a set of fuzzy rules based on its attributes. To build a fuzzy classification system, the most difficult task is to find a set of fuzzy rules pertaining to the specific classification problem [24]. To extract fuzzy rules, a neural net was proposed in several studies [25–27]. On the other hand, the decision tree induction method was used in [28–30]. In [30], a fuzzy decision tree approach was proposed, which can overcome the overfitting problem without pruning and can construct soft decision trees from large datasets. However, these methods were found to be suboptimal in solving certain types of problems [24]. In [13,24], a genetic algorithm for generating fuzzy rules was described. It was noted that they are very robust due to the global searching.

Fuzzy classification has practical applications in various fields. For instance, in [31], a fuzzy rule-based system for classification of diabetes was used. Authors in [32,33] have applied fuzzy theory in managing energy for electric vehicles. In [34], problems of a product processing plant related to the discovery of intrusions in a computer network were solved with use of a fuzzy classifier. In our study, we use fuzziness as a concept of partial patterns.

Both traditional and fuzzy approaches provide for finding both patterns covering the target class and patterns covering the opposite class. The methods for finding such patterns do not differ; in this sense, such approaches are symmetrical: the composition of patterns does not change from replacing the target class with the opposite one. At the same time, there is a significant difference between the requirement for maximum coverage and the requirement for purity of patterns.

When analyzing real data, pure patterns may be ineffective, and the concept of a pattern is extended to fuzzy patterns that cover some of the negative objects. This expansion occurs by relaxing the "empty intersection with negative objects" constraint. Thus, the aim of our study is to construct a classification model based on LAD principles, which does not impose a strict restriction nor relaxed constraint on the pattern coverage of the opposite class observations. Our model converts such a restriction (purity restriction) into an additional criterion. We formulate the pattern search problem as a two-criteria optimization problem: the maximum of covered observations of a certain class with the minimum of covered observations of the opposite class. Thus, our model has two competing criteria of the same scale, and the essence of solving the problem comes down to finding a balance between maximum coverage and purity of patterns. For this purpose, in this paper, we study the use of a multi-criteria genetic algorithm to search for Pareto-optimal fuzzy patterns. Our comparative results on medical test problems are not inferior to the results of commonly used machine learning algorithms in terms of accuracy.

The rest of the paper is organized as follows. In Section 2, we describe known and new methods implemented in our research. We provide the basic concepts of logical analysis of data (Section 2.1), an approach to formation of logical patterns (Section 2.2), a two-criteria optimization model for solving the pattern search problem, concepts of the evolutionary algorithm NSGA-II (Non-dominated Sorting Genetic Algorithm-II), developed to solve the multi-criteria optimization problem (Section 2.4). In Section 2.5, we discover the ability of evolutionary algorithms to solve the problem of generating logical patterns and describe modifications of the NSGA-II (Section 2.5). In Section 3, we present the results of solving two applied classification problems. In Sections 4 and 5, we discuss and shortly summarize the work.

2. An Evolutionary Algorithm for Pattern Generation

Several approaches which resemble in certain respects the general classification methodology of LAD can be distinguished [35]. For instance, in [36], a DNF learning technique was presented that captures certain aspects of LAD. Some machine learning approaches based on production or implication rules, derived from decision trees, such as C4.5 rules [37], or based on Rough Set theory, such as the Rough Set Exploration System [38]. The authors of [39] proposed the concept of emerging patterns in which the only admissible patterns are monotonically non-decreasing. The subgroup discovery algorithm described in [40] maximizes a measure of the coverage of patterns, which is discounted by their coverage of the opposite class. The algorithm presented in [35] maximizes the coverage of patterns while limiting their coverage of the opposite class. In [35], the authors introduced the concept of fuzzy patterns, considered in our paper.

2.1. Main Stages of Logical Analysis of Data

LAD is a data analysis methodology that integrates ideas and concepts from topics, such as optimization, combinatorics, and Boolean function theory [10,41]. The primary purpose of logical analysis of data is to identify functional logical patterns hidden in the data, suggesting the following stages [2,41].

Stage 1 (Binarization). Since the LAD relies on the apparatus of Boolean functions [41], this imposes a restriction on the analyzed data, namely, it requires the binary values of the features of objects. Naturally, in most real-life situations, the input data are not necessarily binary [2,12], and in general may not be numerical. It should be noted that in most cases, effective binarization [41] leads to the loss of some information [2,6].

Stage 2 (Feature extraction). The feature description of objects may contain redundant features, experimental noise as well as artifacts generated or associated with the binarization procedure [42,43]. Therefore, it is necessary to choose some reference set of features for further consideration.

Stage 3 (Pattern generation). Pattern generation is the central procedure for logical analysis of data [2]. At this stage, it is necessary to generate various patterns covering different areas of the feature space. However, these patterns should be of a sufficient level of quality, expressed in the requirements for the parameters of the pattern (for example, complexity or degree). To implement this stage, a specific criterion is selected as well as an optimization algorithm for constructing patterns that is relevant to the data under consideration [44].

Stage 4 (Constructing the classifier). When the patterns are formed, the classification of the new observation is carried out in the following way. An observation that satisfies the requirements of at least one positive pattern and does not satisfy the conditions of any negative patterns is classified as positive. The definition of belonging to a negative class is formulated similarly. In addition, it is required to determine how a decision will be made regarding controversial objects, for example, by voting on the generated patterns. The set of patterns formed at the previous step usually turns out to be too large and redundant for constructing a classifier, which leads to the problem of choosing a representative limited subset of patterns [2], such that it will provide a level of classification accuracy compared to using the complete set of rules. In addition, a decrease in the number of patterns makes it possible to increase the interpretability of the resulting classifier in the subject area [45].

Stage 5 (Validation). The last step of the LAD is not special and is inherent in other data mining methods. The degree of conformity of the model to the initial data should be assessed, and its practical value should be confirmed. In applied problems, the initial data reflect the complexity and variety of real processes and phenomena.

Stage 4 works fine on "ideal" data which means: reasonable amount of homogeneous data with no errors, no outliers (standalone observations which are very far from other ones), no gaps or inconsistency in data. When processing real data, we must take into account several issues [46]. Features may be heterogeneous (of different types and measured on different scales).

Data can be presented in a more complex form than a standard matrix of object features, for example, images, texts, or audio. Various data preprocessing methods are used to extract features. Another type of object description consists in pairwise comparison of objects instead of isolating and describing their features (featureless recognition [47]).

The number of objects may be significantly less than the number of features (data insufficiency). In systems with automatic collection and accumulation of data, the opposite problem arises (data redundancy). There are so much data that conventional methods process them exceptionally slowly. In addition, big amounts of data pose the problem of efficient storage.

The values of the features and the target variable (the label of class in the training sample) can be missed (gaps in data) or measured with errors (inaccuracy in data). Gross errors lead to the appearance of rare but large deviations (outliers). Data may be inconsistent which means that objects with the same feature description belong to different classes as a result of data inaccuracy.

Inconsistency and inaccuracy in data dramatically reduce the effectiveness of approaches based on homogeneous patterns. We have to apply fuzzy patterns in which a certain number of observations of the opposite class are allowed. At the same time, it is difficult to determine this threshold, since the level of data noise is usually unknown.

2.2. Formation of Logical Patterns

We restrict ourselves to considering the case of two classes: K^+ and K^-. Objects of K^+ class will be called positive sampling points, and objects of K^- class will be called negative. In addition, we assume that objects $X \in K^+ \cup K^-$ are described by k binary features, that is $x_i^{(j)} \in \{0,1\}$ $\forall i, j$, where j is the object and i is the index of feature, $j = \overline{1,k}$.

LAD uses terms that are conjunctions of some literals (binary features x_i or their negations $1 - x_i$). We will say that a term C covers an object X if $C(X) = 1$. A logical positive pattern (or simply a pattern) is a term that covers positive objects and does not cover negative objects (or covers a limited number of negative objects). The concept of a negative pattern is introduced in a similar way.

Choose an object $a \in K^+$, assuming that $a = (a_1, \ldots, a_k)$ is the vector of feature values of this object. Denote by P_a a pattern covering the point a. The pattern is a set of feature values, which are fixed and equal for all the objects covered by the pattern. To distinguish fixed and unfixed features in pattern P_a, we introduce binary variables $Y^{(a)} = \left\{y_1^{(a)}, \ldots, y_k^{(a)}\right\}$ [48,49] as follows:

$$y_j^{(a)} = \begin{cases} 1, & \text{if } j\text{th feature is fixed in } P_a, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

A point $b \in K^+$ will be covered by a pattern P_a only if $y_j^{(a)} = 0 \;\; \forall j : \; b_j \neq a_j$. On the other hand, some point $c \in K^+$ will not be covered by the pattern P_a if $y_j = 1$ for at least one $j \in \{\overline{1,k}\}$ for which $c_j \neq a_j$.

It should be noted that any point $Y^{(a)} = \{y_1^{(a)}, \ldots, y_k^{(a)}\}$ corresponds to the subcube in the space of binary features $X = \{x_1, , \ldots, x_k\}$, which includes an object a.

It is natural to assume that the pattern can cover only a part of the observations from K^+. The more observations of positive class the pattern covers in comparison with observations of another type, the more informative it is [50]. Negative observation coverage is a pattern error.

Denote pattern P_a as a binary function of an object b: $P_a(b) = 1$ if object b is covered by pattern P_a, and 0 otherwise.

Noted constraints establish the minimum allowable clearance between the two classes. To improve the reliability of the classification, namely, its robustness to errors on class boundaries, constraints can be strengthened by increasing the value on the right side of the inequality.

Let us introduce the following notation: $Cov^+(P_a)$ is number of observations from K^+ for which the condition $P_a(b) = 1, \;\; b \in K^+$ is satisfied; $Cov^-(P_a)$ is number of observations from K^- for which the condition $P_a(c) = 1, \;\; c \in K^-$ is satisfied.

The pattern P_a is called "pure" if $Cov^-(P_a) = 0$. If $Cov^-(P_a) > 0$, then the pattern P_a is called "fuzzy" [35]. Obviously, among the pure patterns, the most valuable are the patterns with a large number of covered positive observations $Cov^+(P_a)$.

The number of covered positive observations $Cov^+(P_a)$ can be expressed as follows [2]:

$$Cov^+(P_a) = \sum_{b \in K^+} \prod_{j=\overline{1,k},\; b_j \neq a_j} (1 - y_j^{(a)}). \tag{2}$$

The condition that the positive pattern P_a should not cover any point of the negative class, which means the search for a pure pattern requires that for each observation $c \in K^-$, the variable $y_j^{(a)}$ takes the value 1 for at least one j for which $c_j \neq a_j$. Thus, a pure pattern is a solution to the problem of conditional Boolean optimization [2]:

$$\begin{gathered} \sum_{b \in K^+} \prod_{j=\overline{1,k},\; b_j \neq a_j} \left(1 - y_j^{(a)}\right) \rightarrow \max, \\ \sum_{j=\overline{1,k},\; c_j \neq a_j} y_j^{(a)} \geq 1 \quad \forall c \in K^-. \end{gathered} \tag{3}$$

Noted constraints establish the minimum allowable clearance between the two classes. To improve the reliability of the classification, namely, its robustness to errors on class boundaries, constraints can be strengthened by increasing the value on the right side of the inequality.

Since the properties of positive and negative patterns are completely symmetric, the procedure for finding negative patterns is similar.

2.3. Proposed Optimization Model

From the point of view of classification accuracy, pure patterns are preferable [10]. However, in the case of incomplete or inaccurate data, such patterns will have small coverage, which means, for many applications, the rejection of the search for pure patterns in favor of partial.

For the case of partial patterns, the constraint of the optimization problem $Cov^{-}(P_a) = 0$ transforms into the second objective function, which leads to the optimization problem simultaneously according to two criteria:

$$Cov^{+}(P_a) \to \max \quad \text{and} \quad Cov^{-}(P_a) \to \min\,. \tag{4}$$

The least suitable are those patterns that either cover too few observations or cover positive and negative observations in approximately the same proportion. The contradictions between these conflicting criteria can be resolved by transferring the second objective function to the category of constraints through establishing a certain admissible number of covered negative observations. In addition, multi-criteria optimization methods [51] can be applied, which construct an approximation of the Pareto front [52].

In addition, when searching for patterns, it is worth considering the degree of the pattern—the number of fixed signs of this pattern. It is easy to establish that there is an inverse dependence between the pattern degree and the number of covered observations (both positive and negative), so the pattern degree should not be too large [53].

The search for simpler patterns has well-founded prerequisites. Firstly, such patterns are better interpreted and understandable during decision-making. Secondly, it is often believed that simpler patterns have better generalization ability, and their use leads to better recognition accuracy [53]. The use of simple and short patterns leads to a decrease in the number of uncovered positive observations, but at the same time, shorter patterns can increase the number of covered negative observations. A natural way to reduce the number of false positives is to form more selective observations. This is achieved by reducing the size of the pattern determining subcube [48,54].

2.4. Algorithm NSGA-II

The NSGA-II [55] refers to evolutionary algorithms developed for solving the multi-criteria optimization problem that implements the approximation of the Pareto set. It is based on three key components: fast non-dominated sorting, estimation of the solutions location density, and crowded-comparison operator. In the original algorithm, solutions are encoded as vectors of real numbers, and the objective functions are assumed to be real-valued.

2.4.1. Fast Non-Dominated Sorting Approach

The basis of multi-criteria optimization is the selection of Pareto fronts [52] of different levels in the population of solutions. According to the intuitive approach, to isolate the non-dominated front, it is necessary to compare each solution with any other in the population for determining a set of non-dominated solutions.

For problem (4), the solutions are patterns, and a solution P_i is non-dominated in a population of N solutions $P_1, \ldots, P_N$ if $\nexists j \in \overline{1, N}$:

$$\left(Cov^{+}(P_j) \geq Cov^{+}(P_i) \text{ and } Cov^{-}(P_j) < Cov^{-}(P_i)\right)$$

$$\text{or } \left(Cov^{+}(P_j) > Cov^{+}(P_i) \text{ and } Cov^{-}(P_j) \leq Cov^{-}(P_i)\right).$$

After defining the first front, it is necessary to exclude representatives of this front from consideration and re-define the non-dominated front. The procedure is repeated until each solution is attributed to some front.

A more computationally efficient approach assumes for each solution P_j to keep track of the number of solutions n_{P_j} that dominate P_j, as well as a set of solutions S_{P_j} dominated by P_j.

Thus, all decisions in the first non-dominated front will have a dominance number equal to zero. For every solution P_j satisfying $n_{P_j} = 0$, we visit each member q of its set S_{P_j} and decrease its dominance number n_q by 1. Moreover, if for any member, the dominance number becomes zero, we put it in a separate list Q. Bypassing all the solutions of the first front, we find that the list Q contains solutions belonging to the second front. We repeat the procedure for these solutions similarly, looking through the dominated sets of solutions and reducing their dominance number until we identify all the fronts.

2.4.2. Diversity Maintenance

One of the problems of evolutionary algorithms is maintaining the diversity of the population [46,56,57]. Eremeev in [58] described the mutation genetic operator as the essential procedure that guarantees population diversity. Along with the convergence to the Pareto optimal set to solve the problem of multi-criteria optimization, the evolutionary algorithm must also support a good distribution of solutions, preferably evenly covering as much of the optimal front as possible [59–62].

Early versions of the NSGA used the well-known fitness-sharing approach to prevent the concentration of solutions in specific areas of the search space and maintain stable population diversity. However, the proximity parameter ς_{share} has a significant effect on the efficiency of maintaining a wide distribution of the population. This parameter determines the degree of redistribution of fitness between individuals [55] and is directly related to the distance metric chosen to calculate the measure of proximity between two members of the population. The parameter ς_{share} denotes the largest distance within which any two solutions share each other's suitability. The user usually sets this parameter, which entails apparent difficulties in making a reasonable choice.

In the NSGA-II [55], a different approach is used based on the crowded comparison. Its indisputable advantage is the absence of parameters set by the user. The critical components of the approach are the density estimate and the crowded-comparison operator.

To estimate the density of solutions concerning the chosen solution, along each direction in the criteria space, two nearest solutions are found on both sides of the chosen solution. The distance between them is determined as the difference between the values of a different criterion. An estimate of the density of solutions near the selected point will be the average of the calculated distances, called the crowding distance. From the geometric point of view, it is possible to estimate the density by calculating the perimeter of the hypercube formed by the nearest neighboring solutions as vertices.

Figure 1 shows a graphical interpretation of the above approach for the case of our two objective functions (4). Solutions denoted as p_{i+1} and $i - 1$, which are nearest to the ith solutions and belong to the first front (filled dots), represent the vertices of the outlined rectangle (dotted line). Crowding distance, in this case, can be defined as the average length of the edges of the rectangle.

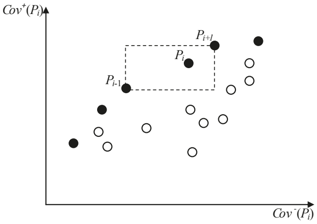


Figure 1. Density estimation of solutions belonging to the front for problem (4).

Calculating the crowding distance requires sorting the individuals in the population according to each objective function in ascending order of the value of this objective function. For individuals with the boundary value of the objective function (maximum or minimum), the crowding distance is assigned equal to infinity. All other intermediate solutions are assigned a distance value equal to the absolute value of the difference between the values of the functions of two neighboring solutions. This calculation continues for all objective functions. The final value of the crowding distance is calculated as the average of the individual values of the distances corresponding to each objective function. Pre-normalization of objective functions is recommended.

After all individuals in the population have been assigned to an estimate of the crowding distance, we can compare the solutions in terms of their degree of closeness to other solutions. A smaller value for the crowding distance indicates a higher density of solutions relative to the selected point. The density estimate is used in the crowded-comparison operator described below.

Crowded-Comparison Operator ($\prec_n$) directs the evolutionary process of population transformation towards a fairly uniform distribution along the Pareto front.

We assume that each individual in the population has two attributes:

1. Rank of dominance (front rank) i_{rank};
2. Crowding distance $i_{distance}$.

Then, the partial ordering is defined as follows. Individual i is preferred over individual j if the following conditions are met:

$$i \prec_n j \Longleftrightarrow (i_{rank} < j_{rank}) \bigvee \left(i_{rank} = j_{rank} \bigwedge i_{distance} > j_{distance}\right). \quad (5)$$

Such ordering means that we prefer a solution with a lower (close to the first, which means the optimal front) rank between two solutions with different ranks. Otherwise, if both solutions are located in the same front, we prefer a solution located in areas where solutions are less crowded.

2.4.3. Basic Procedure of the NSGA-II

In this case, solving linear optimization problem is rather simple. Initially, the parent population P_0 is randomly created and sorted based on the dominance principle. Thus, the greater the suitability of an individual, the lower the value of its rank. Then, traditional genetic operators are applied: binary tournament selection, crossover, mutation, creating a child population Q_0 of a specified size N. Since elitism is introduced by comparing the current child population with the parent population, the procedure differs for the first generation from the repeated one.

Let the tth iteration of the algorithm be executed, at which the parent population P_t generated the child population Q_t. A joint population $R_t = P_t \cup Q_t$ is then sorted according to the dominance principle. Thus, decisions that belong to the first front $\mathcal{F}_1$ and are not dominated should have a better chance of moving to the next generation population, which is ensured by the implementation of the elitism principle. If the first front $\mathcal{F}_1$ includes less than N members, then all members of this front move to the next population P_{t+1}. The remaining members of the population P_{t+1} are selected from subsequent fronts in the order of their ranking. That is, the front $\mathcal{F}_2$ is included in the new parent population P_{t+1}, then the front $\mathcal{F}_3$, and so on. This procedure continues until the inclusion of the next front leads to an excess of the population size N. Let the front $\mathcal{F}_l$ no longer be included in the new population as a whole. To select members of the front $\mathcal{F}_l$ who will be included in the next generation, we sort the solutions of this front using the crowded-comparison operator and select the best solutions to supplement the population P_{t+1}. Now, a new population P_{t+1} of size N can be used to apply selection, crossover, and mutation operators. It is important to note that the tournament selection binary operator is still used, but now, it is based on the crowded-comparison operator $\prec_n$, to use which, in addition to determining the rank, it is necessary to calculate the crowding distance for each member of the population P_{t+1}.

The schematic procedure of the NSGA-II algorithm originally proposed in [55] is shown in Figure 2.

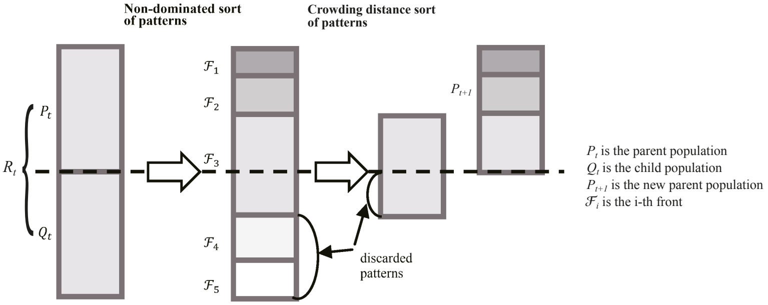

Non-dominated sort of patterns
Crowding distance sort of patterns
$\mathcal{F}_1$
$\mathcal{F}_2$
$\mathcal{F}_3$
$\mathcal{F}_4$
$\mathcal{F}_5$
P_t
R_t
Q_t
P_{t+1}
discarded patterns
P_t is the parent population
Q_t is the child population
P_{t+1} is the new parent population
$\mathcal{F}_i$ is the i-th front

Figure 2. Next-generation transition procedure.

Thus, a successful distribution of solutions within one front is realized by using the crowded-comparison procedure, which is also used in the individuals selection. Since the solutions compete with their crowding distances (a measure of the solution's density in a neighborhood), the algorithm does not require any additional parameter determining the size of niches in the search space. The proposed crowding distance is calculated in the function space, but it can be implemented in the parameter space if necessary.

2.5. Our Approach: An Evolutionary Algorithm for Pattern Generation

The pattern P_a is determined by baseline observation $a = \{x_1, x_2, \ldots, x_p\}$ and values of control variables $Y = \left\{y_1^{(a)}, \ldots, y_p^{(a)}\right\}$ that are binary values. Thus, the solution to the problem of pattern generation is a set of points in the space of control variables that approximate the Pareto front for a given base observation.

The posed problem of finding logical patterns determines the binary representation of solutions in the form of binary strings, rather than real variables, as was postulated in the original NSGA-II. The binary representation of the solution also determined the list of available crossover and mutation operators, among which uniform crossover and mutation by gene inversion were chosen [63,64].

In addition, the crowding distance in space "the number of covered observations of the base class" requires a different interpretation—"the number of covered observations of the class other than the base". Firstly, uniform coverage of the Pareto front is not required, since a preferable area is an area with a smaller scope of observations of the opposite class. Figure 3 illustrates this position: points of one Pareto front are colored according to their preference, with white color meaning less preference.

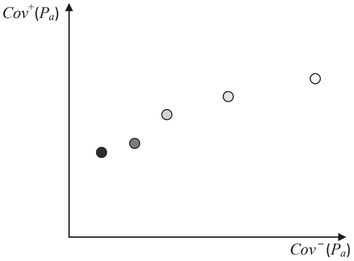


Figure 3. Illustration of a typical first Pareto front for logical patterns.

Taking into account these preconditions, the crowding distance was replaced by one of the heuristic definitions of informativity [3]:

$$I = \sqrt{Cov^{+}(P_a)} - \sqrt{Cov^{-}(P_a)}. \tag{6}$$

Similar to the crowding distance, the more preferable the individual, the greater the informative value.

Special attention is paid to the formation of the initial set of individuals. The usual approach is a uniform discrete distribution for values 0 and 1. However, based on the nature of the problem, an equal probability of dropping out 0 and 1 will mean, on average, the fixation of half of the features in the initial population, which gives rise to overly selective patterns. Therefore, the discrete distribution of values in the original population is defined differently: dropout 1 with probability p, and dropout 0 with probability $1 - p$. The value p will be the hyperparameter of the genetic algorithm.

Figure 4 shows a diagram of the developed approach to construct a classifier using the NSGA-II algorithm for pattern generation. The NSGA-II algorithm is run independently for each baseline observation. The launch result is a set of patterns of the first front. The sets of patterns obtained for each baseline observation were combined into one complete set of patterns, which can be reduced using the selection procedure [2,65]. When recognizing control (or new) observations, the decision about the class is made by balanced voting of patterns on the observation under consideration [8].

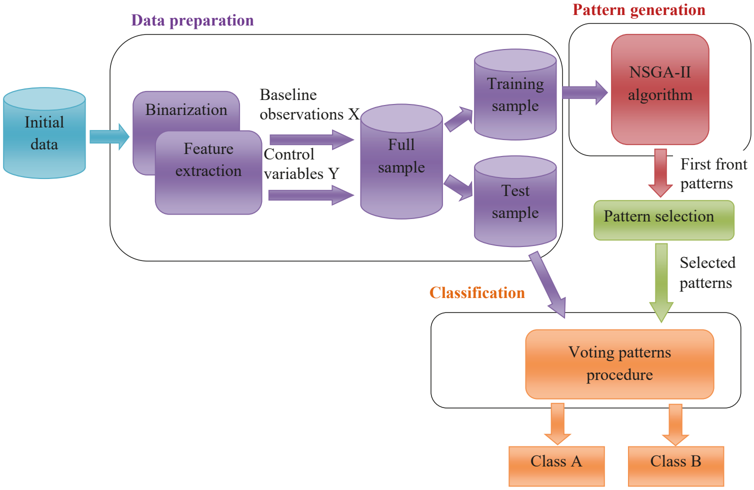


Figure 4. Scheme for constructing a classifier using the algorithm NSGA-II.

A detailed description of pattern generation procedure from Figure 4 is presented in a form of pseudocode (Algorithms 1 and 2).

Algorithm 1 Fast-non-dominated-sorting

```
Require: Evaluated population $P = \{P_1, \dots, P_N\}$, number of solutions in population $N$.
for each $P_i$, $i \in \{\overline{1,N}\}$ do
    for each $P_j$, $j \in \{\overline{1,N}\}$ do
        if $P_i$ dominates $P_j$ ($P_i \prec P_j$) then
            increase the set of solutions which the current solution dominates:
            $S_{P}i \leftarrow S_{P}i \cup \{P_j\}$
        else
            if $P_j$ dominates $P_i$ ($P_j \prec P_i$) then
                increase the number of solutions which dominate the current solution:
                $n_{P}i \leftarrow n_{P}i + 1$
            end if
        end if
    end for
    if no solution dominates $P_i$, $(n_{P}i = 0)$ then
        $P_i$ is a member of the first front: $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{P_i\}$
    end if
end for
$t \leftarrow 1$
while $\mathcal{F}_t \neq \emptyset$ do
    $\mathcal{H} \leftarrow \emptyset$
    for each $P_i \in \mathcal{F}_t$ do
        for each $P_j \in S_{P}i$ do
            $n_{P}j \leftarrow n_{P}j - 1$
            if $n_{P}j = 0$ then
                $\mathcal{H} \leftarrow \mathcal{H} \cup \{P_j\}$
            end if
        end for
    end for
    $t \leftarrow t + 1$
    $\mathcal{F}_t \leftarrow \mathcal{H}$
end while
return a list of the non-dominated fronts $\mathcal{F}$
```

NSGA-II is one of the popular multiobjective optimization algorithms. It is usually assumed that each individual in the population represents a separate solution to the problem. Thus, the population is a set of non-dominated solutions, from which one or more solutions can then be selected, depending on which criteria are given the highest priority. A distinctive feature of the proposed algorithm is, among other things, that the solution to the problem is the entire population, in which individual members represent individual patterns.

Algorithm 2 An evolutionary algorithm for pattern generation

```
Require: The set of baseline observations $X$ and values of control variables $Y$
Create a random parent population $P_0$ of size $N$ from $X$ and $Y$
$\mathcal{F} \leftarrow Fast - nondominated - sort(P_0)$
Create a child population of size $N$ using selection, crossover and mutation procedures $Q_0 \leftarrow Child(P_0)$
$t \leftarrow 0$
while termination criteria not met do
    Combine parent and child populations $R_t \leftarrow P_t \cup Q_t$. The size of population $R_t$ is $2N$
    $\mathcal{F} \leftarrow Fast - nondominated - sort(R_t)$
    while $|P_{t+1}| \geq N$ do
        $I \leftarrow \sqrt{Cov^+(P_{t+1})} - \sqrt{Cov^-(P_{t+1})}$
        $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$
    end while
    Sort $P_{t+1}$ in descending order
    $P_{t+1} \leftarrow thefirstNelementsfromP_{t+1}$
    Create a child population using selection, crossover and mutation procedures $Q_{t+1} \leftarrow Child(P_{t+1})$
    $t \leftarrow t + 1$
end while
return $P_{t+1}$
```

3. Application of the Proposed Method to Problems in Healthcare

The proposed approach relates to interpretable machine learning methods. Therefore, experimental studies were carried out on problems in which the interpretability of the recognition model and the possibility of a clear explanation of the solution proposed by the classifier are of great importance. The results are shown on two datasets: breast cancer diagnosis (the dataset from the UCI repository [66]) and predicting complications of myocardial infarction (regional data [67]).

3.1. Breast Cancer Diagnosis

The problem of diagnosing breast cancer on a sample collected in Wisconsin is considered (Breast Cancer Wisconsin, BCW) [66]. Since the attributes in the data take on numeric (integer) values, their binarization is necessary, that is, the transition to new binary features. Threshold-based binarization is used. Based on the original value x, a new binary variable x_t can be constructed as follows:

$$x_t = \begin{cases} 1, & \text{if } x \geq t, \\ 0, & \text{if } x < t, \end{cases} \tag{7}$$

where t is a cut point (threshold value).

As a result of executing the binarization procedure, 72 binary attributes were obtained from 9 initial attributes. The original dataset is divided in the ratio of 70% and 30% into training and test samples (in a random way), which results in 478 and 205 observations, respectively. To search for logical patterns, objects of the training sample (both positive and negative classes) consistently act as a baseline observation. For each baseline observation, the NSGA-II algorithm is run independently with the following parameters:

1. Parent population size: 20;
2. Descendant population size: 20;
3. Number of generations: 10;
4. Probability of mutation (for each decision variable): 5%;
5. Cross type: uniform.

The final set of patterns includes patterns of the first front from the last population of solutions. The first front can contain one or several patterns. Thus, the final set of patterns is completed based on covering the observations of the training sample. Therefore, each observation will be covered by at least one pattern from this set.

A series of experiments are carried out with a different probability of dropping one when initializing control variables in the first population of the genetic algorithm. The following probabilities were used: $p \in \{0.5, 0.3, 0.1\}$. A complete set of patterns was independently constructed for each probability. This set included the patterns of the first fronts obtained using the NSGA-II algorithm with the sequential acceptance of the observations of the training sample as a baseline observation. The resulting complete set of patterns are shown in Figure 5 in the coordinates of the number of covered observations of its own and the opposite class.

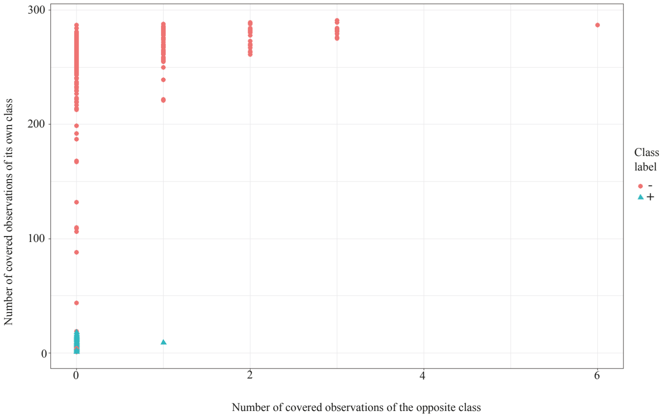


Figure 5. Aggregate scatter diagram of the detected patterns with $p = 0.5$.

Figure 6 shows the resulting full set of patterns for the value $p = 0.3$.

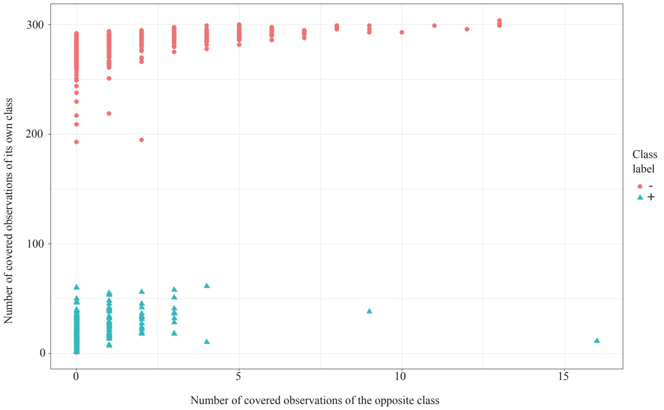


Figure 6. Aggregate scatter diagram of the detected patterns with $p = 0.3$.

Figure 7 shows the resulting full set of patterns for the value $p = 0.1$.

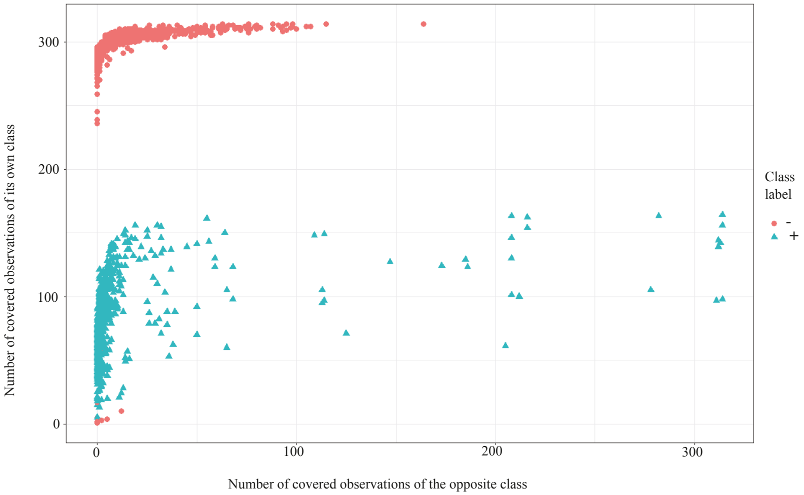


Figure 7. Aggregate scatter diagram of the detected patterns with $p = 0.1$.

Table 1 shows the distribution of patterns by power. Power is understood as the number of covered observations of its class included in the training sample. Data on the aggregate set of patterns are given without considering their class affiliation.

Table 1. Distribution of patterns in terms of power.

	Probability of Dropping 1 on Initialization		
The Number of Covered Observations of The Same Class on The Training Sample	***p* = 0.5**	***p* = 0.3**	***p* = 0.1**
1	1240	200	20
from 1 to 5	394	113	4
from 5 to 100	106	268	545
from 100 to 300	465	974	1770
above 300	0	3	962
Total number of patterns detected	2205	1558	3301

In the case of equally probable dropping of zero and one during initialization, a large number of trivial patterns are found, that is, patterns that cover only the basic observation and do not cover any more observations. These patterns are overly selective. With a decrease in the probability of dropping one during initialization, the selectivity of the obtained patterns decreases, thereby increasing their coverage. However, increased coverage affects the accuracy of initiated patterns, i.e., the number of observations from the opposite class.

The results of the classification of test observations compared to actual values are shown in Table 2. Symbols "+" and "−" denote the class labels introduced above. The symbol "?" means the impossibility of classification because none of the patterns is valid for observation.

Table 2. Confusion matrix for BCW problem.

***p* = 0.5**			***p* = 0.3**			***p* = 0.1**		
	Original Class			**Original Class**			**Original Class**	
Classifier Prediction	**—**	**+**	**Classifier Prediction**	**—**	**+**	**Classifier Prediction**	**—**	**+**
—	126	4	—	128	8	—	128	12
+	0	37	+	2	65	+	2	63
?	4	34	?	0	2	?	—	—
Accuracy	0.888		Accuracy	0.946		Accuracy	0.932	

Thus, a significant influence of the probability of dropping one is established during the initialization of control variables in the first population of the multi-criteria genetic algorithm. This parameter affects the selectivity of the detected patterns. The equiprobable dropping of zero and one entails a significant proportion of baseline observations, for which only patterns that cover only the baseline observation and no other training observations are found.

3.2. Predicting the Complication of Myocardial Infarction

The problem of predicting the development of complications of myocardial infarction is considered [67]. Datasets are often significantly asymmetric: one of the classes significantly outnumbers the other in terms of the number of objects. In our case, a sample contains data on 1700 cases with an uneven division into classes: 170 cases with complications, and 1530 cases without. Each case in the initial sample is described by 127 attributes containing information about the history of each patient, the clinical picture of the myocardial infarction, electrographic and laboratory parameters, drug therapy, and the characteristics of the course of the disease in the first days of myocardial infarction. Data include the following types: textual data, integer values, rank scale values with known range, real and binary values. A more detailed description of the attributes is given in Table 3.

The problem of predicting atrial fibrillation (AF) is solved, assuming that it is described by the variable 116 (target variable, 1—the occurrence of atrial fibrillation, 0—its absence). According to the nature of the attributes, variables 94, 95, 97, 98, 104, 105, 107, 108 were excluded since these values could be obtained after the occurrence of complications [67]. The exclusion of these attributes significantly complicates the forecasting task. Thus, the number of predictors from the initial data was 107 variables.

The properties of the data used are as follows:

1. A significant number of missing values;
2. "Asymmetry" of the sample: the number of patients with atrial fibrillation is only about 10% of the total;
3. The presence of different types of attributes;
4. The use of homogeneous patterns leads to overfitting (the formation of patterns with a large number of conditions and a very small coverage).

Two approaches to data processing were implemented when constructing the classifier, as described below.

Table 3. Description of sample attributes for myocardial infarction problem.

Attribute Number	Description	Value Type
1	Full name of the patient (meta attribute)	text
2	Age	integer
3	Gender	binary
4–34	The presence or absence of various pathologies in the cardiovascular, endocrine and respiratory systems before the development of myocardial infarction	rank, binary
35–38	Blood pressure according to the data of the cardiology team and the admission department	integer
39–44	Complications arising at the time of transportation of the patient to the clinic or at the time of hospitalization	binary
45–49	Depth and localization of cardiac muscle necrosis	rank, binary
50–75	ECG characteristics at the time of admission of the patient to the intensive care unit	binary
76–82	The use of drugs during fibrinolytic therapy	binary
83–86	Electrolyte shifts in the blood	binary, real
87–92	Laboratory characteristics of blood	real, rank
93–115	The course of the disease in the first days of myocardial infarction	rank, binary
116–127	The occurrence of complications of myocardial infarction or the outcome of this disease	rank, binary

3.2.1. First Approach to Data Preparation (Complete Data and Handling of Missing Values)

The data contained a significant number of missing values. Columns with more than 100 missing values were excluded (47 variables). The remaining variables contained rank scale values and binary values. Missing values in columns containing 100 or less gaps were filled with the mode value. Thus, the prepared dataset contained 60 variables (not including the target).

The binarization for the values presented on the rank scale was performed based on the following rules. Based on the original variable x, a new binary variable x_t was constructed as follows:

$$x_t = \begin{cases} 1, & \text{if } x = r, \\ 0, & \text{if } x \neq r, \end{cases} \tag{8}$$

where r is the value of the original variable $x, r \in R$, such that R is a known set of all possible values of a variable x. As a result of the binarization procedure, the total number of binary variables was 119.

The original dataset is divided in a ratio of 70% and 30% into training and test samples, that is 1190 and 510 observations, respectively. To search for patterns, the training sample's objects (both positive and negative classes) consistently act as a baseline observation. For each baseline observation, the NSGA-II algorithm is run independently with the following parameters:

1. Parent population size: 20;
2. Descendant population size: 20;
3. Number of generations: 10;
4. Probability of dropping one on initialization: 0.05;
5. Probability of mutation (for each decision variable): 0.5%;
6. Crossing type: uniform.

More resources are allocated for seeking positive class observations. The sizes of the parent and descendant populations are 50, and the number of generations was 100. As a result, a complete set of patterns was obtained, containing 1961 positive class patterns and 6148 negative class patterns. A classifier is built that decides on a new observation by a simple voting. The results of the classification of the test observations compared to the actual values are shown in Table 4. Symbols "+" and "−" denote the class labels introduced above.

Table 4. Confusion matrix for AF problem (first approach).

	Original Class	
Classifier Prediction	**−**	**+**
−	347	31
+	109	23
Accuracy	0.7255	

Thus, the classifier arising from data for which missing values were processed did not show an acceptable result of classifying positive class objects, which may be caused by deleting essential data due to missing values processing. Since logical patterns are usable for classifying data with missing values, a different approach to data preparation was used, which is described below.

3.2.2. Another Approach to Data Preparation (Reduced Sampling without Processing Missing Values)

The second approach to data preparation is to store the missing attribute values as in the original sample. The class imbalance problem is also solved by randomly selecting a subset of objects of the prevailing class, equal in cardinality to the set of objects of another class.

As a result, the original sample was reduced to 338 cases, with an equal number of instances belonging to each class. Observations are described by 106 variables (not including the target). On the reduced sample, 8 variables take either just the same value, or the same value and gaps, so these variables were excluded from consideration. Among the remaining variables, 16 are rank variables, 12 are real, and the other variables are binary. Rank variables are transformed into several new binary variables, whose number is determined by the number of allowed ranks. Based on each real variable, four new binary variables are built. The threshold values of these variables were divided by the range of values of the original variables into four equal parts. As a result of applying the binarization procedure, the total number of variables was 200 (excluding the target variable).

Data were divided into training and test samples in the ratio of 80% and 20%, respectively, which causes 270 and 68 observations, respectively, in the same number of observations of positive and negative classes. For each training set observation, the NSGA-II algorithm is run independently with the following parameters:

1. Parent population size: 20;
2. Descendant population size: 20;
3. Number of generations: 10;
4. Probability of dropping one on initialization: 0.05;
5. Probability of mutation (for each decision variable): 0.5%;
6. Crossing type: uniform.

As a result, the cumulative set of patterns for the first front includes 2259 rules for the positive class and 2403 for the negative class. The classifiers are built based on reduced sets with the selection according to informativeness (I) of revealed patterns. Simple voting of patterns classifies a new observation. If the attribute value fixed in the pattern is unknown in the new observation, we assume that this pattern does not cover this observation. The classification results for different threshold values of informativeness are shown in Table 5.

Table 5. Confusion matrix for AF problem (the second approach).

$I > 0$			$I > 1$			$I > 2$		
	Original Class			**Original Class**			**Original Class**	
Classifier Prediction	**−**	**+**	**Classifier Prediction**	**−**	**+**	**Classifier Prediction**	**−**	**+**
−	25	7	−	25	7	−	25	5
+	9	27	+	9	27	+	9	29
Accuracy	0.7647		Accuracy	0.7647		Accuracy	0.7941	
$I > 0$			$I > 1$			$I > 2$		
	Original Class			**Original Class**			**Original Class**	
Classifier Prediction	**−**	**+**	**Classifier Prediction**	**−**	**+**	**Classifier Prediction**	**−**	**+**
−	23	4	−	23	3	−	30	18
+	11	30	+	11	31	+	4	16
Accuracy	0.7794		Accuracy	0.7941		Accuracy	0.6765	

Thus, the greatest accuracy is achieved when selecting patterns with informativeness greater than 2 or greater than 4. Since the importance of correctly identifying positive class objects (true positive rate or sensitivity) is greater than a negative one (true negative rate or specificity), the best option is to select patterns with informativeness greater than 4, since the accuracy of classifying objects in the positive class is higher. The obtained classification accuracy exceeds the accuracy obtained in [44].

Let us explore some characteristics of the resulting patterns when choosing patterns based on $I \geq 4$. Table 6 shows the number of patterns as well as the degree distribution of these patterns, that is, the number of binary variables fixed in the pattern.

Table 6. Degree of the patterns.

	Amount	Min	1st Quartile	Median	Mean	3rd Quartile	Max
Positive patterns	96	3	6	7	7.802	9	16
Negative patterns	103	6	9	11	11.480	14	19
All patterns	199	3	7	9	9.704	12	19

Figure 8 shows the indicated sets of patterns in the coordinates of the number of covered observations of its class and the opposite class. Figure 8 is given for the training sample.

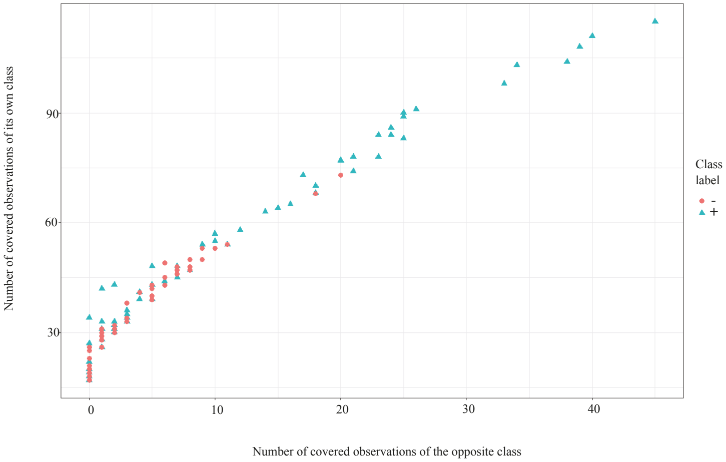


Figure 8. Cumulative scatter diagram of patterns on the training set.

Figure 9 shows the indicated sets of patterns in the coordinates of the number of covered observations of its class and the opposite class for the test sample.

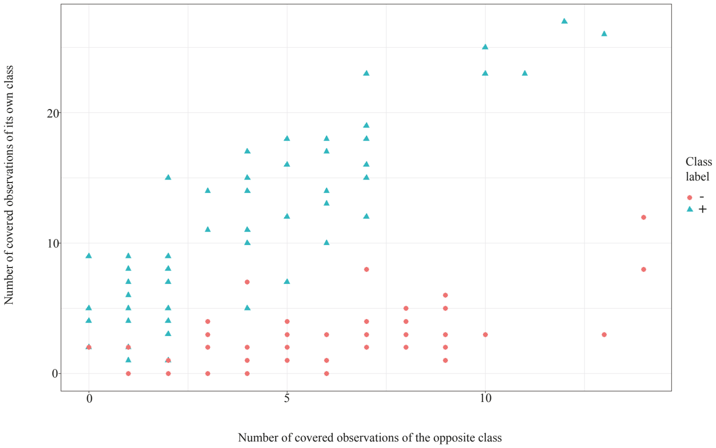


Figure 9. Cumulative scatter diagram of patterns on the test sample.

To evaluate the efficiency of the proposed approach, we made a comparative study with some widely used machine learning classification algorithms. We compared the proposed multi-criteria genetic algorithm (MGA-LAD) with Support Vector Machine (SVM), C4.5 Decision Trees (J48), Random Forest (RF), Multilayer Perceptron (MP), and Simple Logistic Regression (LR) methods. The tests were performed on the following datasets from UCI Machine Learning Repository [66]: Wisconsin breast cancer (BCW), Hepatitis (Hepatitis), Pima Indian diabetes (Pima), Congressional voting (Voting). Results using 10-fold cross-validation are presented in Table 7. Here, the "ML" column contains results for algorithms that give the highest accuracy among the commonly used machine learning algorithms. The best algorithm is given in parentheses. Another approach that has been used for comparison is logical analysis of data (LAD-WEKA in the WEKA package [68]) at various fixed fuzziness values φ (an upper bound on the number of points from another class that is covered by a pattern as a percentage of the total number of points covered by the pattern).

Table 7. Classification accuracy of the compared algorithms.

Dataset	ML	LAD, Fuzziness at Most φ					MGA-LAD
		φ = 0	φ = 0.05	φ = 0.1	φ = 0.15	φ = 0.2	
BCW	0.967 ± 0.01 (RF)	0.951 ± 0.03	0.953 ± 0.02	0.957 ± 0.03	0.966 ± 0.03	0.966 ± 0.03	0.965 ± 0.02
Hepatitis	0.858 ± 0.08 (SVM)	0.819 ± 0.09	0.826 ± 0.09	0.819 ± 0.09	0.806 ± 0.09	0.806 ± 0.08	0.847 ± 0.10
Pima	0.736 ± 0.03 (RF)	0.682 ± 0.02	0.687 ± 0.02	0.691 ± 0.02	0.687 ± 0.02	0.682 ± 0.02	0.749 ± 0.04
Voting	0.961 ± 0.01 (LR)	0.945 ± 0.03	0.952 ± 0.03	0.954 ± 0.03	0.954 ± 0.03	0.949 ± 0.03	0.973 ± 0.03

The value of the fuzziness parameter φ affects the size, coverage, and informativeness of the resulting patterns and ultimately affects the classification accuracy. In LAD, it is necessary to find many *a*-pattern based on different baseline observations. However, for each such pattern, the most appropriate fuzziness value may be different. Using a two-criteria model and the corresponding optimization algorithm allows us to find a set of Pareto optimal patterns without the need to fix the fuzziness value, which expands the possibilities of LAD and can improve the classification accuracy.

4. Discussion

In the previous section, we described the application of the proposed approach to solving practical medical classification problems.

The first dataset describes tissue characteristics to diagnose benign or malignant breast neoplasm. This dataset was studied many times before, for example [69,70], including the approach based on LAD [2]. For this example, we established a significant influence of the introduced hyperparameter of the genetic algorithm—the probability of dropping one during the initialization of the initial population, which is the fixation of the attribute's value in the pattern according to its value in the baseline observation. High values of this probability lead to low classification accuracy due to excessive selectivity of patterns and, accordingly, an increase in the number of objects with a refusal to determine the class membership.

The second dataset is the problem of predicting complications of myocardial infarction-atrial fibrillation. In this case, two approaches are implemented to handle missing values in the data. The first, typical for most data mining algorithms, is a combination of deleting values with missing values and filling them in. In this case, satisfactory classification results were not achieved when a complete dataset with a significantly larger presence of objects of one of the classes was used. In the second approach, the missing values are not preprocessed since the set of logical patterns as a whole has no restrictions in classifying observations with missing values. In addition, we use a reduced sample with an equal number of observations for each of the classes.

Homogeneous patterns in this dataset have small coverage, and using only homogeneous patterns leads to overfitting. Relaxation of homogeneity constraints requires adjusting the threshold (right-hand side of the constraint), which can be difficult since, when solving pattern finding problems, the best balance between coverage and homogeneity for a single pattern can be far from the best for another pattern (based on another baseline observation). The proposed approach simplifies the search for this balance since it considers many Pareto-optimal patterns. This approach prevents overfitting in contrast with using only homogeneous patterns or patterns with a given threshold for homogeneity. At the same time, the accuracy reaches the values obtained using an artificial neural network specially developed for these data [67]. The classification results are also comparable with the results of other works on this topic [71,72].

5. Conclusions

Logical analysis of data is a two-class learning method dealing with features that can be binarized. Logical analysis of data consists of several stages, and the formation of patterns is the most important. Finding pure patterns is a single-criteria optimization problem that consists of finding a pattern that covers as many positive observations as possible and does not cover negative observations. With a more general formulation, which allows making mistakes for a pattern, the problem turns into a search for a compromise between the completeness of the range of observations of some target class and the minimization of the coverage of observations that do not belong to the target class.

In this study, we used an approach to find patterns in the form of an approximation of the Pareto-optimal front and proposed evolutionary algorithms for solving such a problem due to their potential ability to cover the front. We modified the NSGA-II for pattern searching and tested it on several application problems from the repository.

Results of our work enabled us to find more informative patterns in the data, taking into account the coverage of objects of different classes. We compared our modified algorithm with commonly used machine learning algorithms on four classification problems. The results were comparable, and in some cases better than results of classical ML algorithms which do not meet the requirement of the interpretability of the result.

Our experiments discovered a significant influence of the probability of dropping one of the control variables of the initial population in the multi-criteria genetic algorithm. This parameter affects the selectivity of the selected patterns. So, the equal probability of zero and one entails a significant proportion of baseline observations, for which only patterns are found that cover only the baseline observation and no other observation of the training sample.

Since our two-criteria optimization model in a combination with the developed modification of the genetic algorithm does not require the number or ratio of observations of the opposite class to be pre-set. Thus, our approach is a more versatile tool of data analysis in this sense than known methods for the fuzzy patterns generation. The method considered in this paper could be useful for the classification tasks in, for instance, healthcare system, faults diagnosis and any problems in which the interpretability of results are of great importance.

Author Contributions: Conceptualization, I.S.M. and M.A.K.; methodology, I.S.M. and L.A.K.; software, M.A.K.; validation, I.S.M., M.K. and E.M.T.; formal analysis, P.S.S. and L.A.K.; investigation, I.S.M.; resources, I.S.M.; data curation, I.S.M. and M.A.K.; writing—original draft preparation, I.S.M., E.M.T. and L.A.K.; writing—review and editing, L.A.K., E.M.T., A.A.S. and P.S.S.; visualization, I.S.M., M.A.K. and A.A.S.; supervision, A.M.P., L.A.K. and P.S.S.; project administration, L.A.K. and A.M.P.; funding acquisition, L.A.K. and A.M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Ministry of Science and Higher Education of the Russian Federation, State Contract FEFE-2020-0013.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Initial data available at UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original), access date: 13 February 2022; https://archive.ics.uci.edu/ml/datasets/Myocardial+infarction+complications, access date: 13 February 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hammer, P.L. Partially defined boolean functions and cause-effect relationships. In *Proceedings of the International Conference on Multi-Attribute Decision Making Via OR-Based Expert Systems*; University of Passau: Passau, Germany, 1986.
2. Hammer, P.L.; Bonates, T.O. Logical analysis of data: From combinatorial optimization to medical applications. *Ann. Oper. Res.* **2006**, *148*, 203–225. [CrossRef]
3. An, A.; Cercone, N. Rule Quality Measures for Rule Induction Systems: Description and Evaluation. *Comput. Intell.* **2001**, *17*, 409–424. [CrossRef]
4. Bruni, R.; Bianchi, G.; Dolente, C.; Leporelli, C. Logical Analysis of Data as a tool for the analysis of Probabilistic Discrete Choice Behavior. *Comput. Oper. Res.* **2019**, *106*, 191–201. [CrossRef]
5. Han, J.; Kim, N.; Yum, B.; Jeong, M. Pattern selection approaches for the logical analysis of data considering the outliers and the coverage of a pattern. *Expert Syst. Appl.* **2011**, *38*, 13857–13862. [CrossRef]
6. Boros, E.; Hammer, P.L.; Ibaraki, T.; Kogan, A.; Mayoraz, E.; Muchnik, I. An Implementation of Logical Analysis of Data. *IEEE T. Knowl. Data En*. **2000**, *12*, 292–306. [CrossRef]
7. Crama, Y.; Hammer, P.L.; Ibaraki, T. Cause-effect relationships and partially defined Boolean functions. *Ann. Oper. Res.* **1988**, *16*, 299–326. [CrossRef]
8. Lejeune, M.; Lozin, V.; Lozina, I.; Ragab, A.; Yacout, S. Recent advances in the theory and practice of Logical Analysis of Data. *Eur. J. Oper. Res.* **2019**, *275*, 1–15. [CrossRef]
9. Bain, T.; Ávila, J.; Subasi, E.; Subasi, M. Logical analysis of multiclass data with relaxed patterns. *Ann. Oper. Res.* **2020**, *287*, 11–35. [CrossRef]
10. Alexe, G.; Hammer, P.L. Spanned patterns for the logical analysis of data. *Discrete Appl. Math.* **2006**, *154*, 1039–1049. [CrossRef]
11. Guo, C.; Ryoo, H. On Pareto-Optimal Boolean Logical Patterns for Numerical Data. *Appl. Math. Comput.* **2021**, *403*, 126153. [CrossRef]
12. Lejeune, M.A. Pattern-based modeling and solution of probabilistically constrained optimization problems. *Oper. Res.* **2012**, *60*, 1356–1372. [CrossRef]
13. Caserta, M.; Reiners, T. A pool-based pattern generation algorithm for logical analysis of data with automatic fine-tuning. *Eur. J. Oper. Res.* **2016**, *248*, 593–606. [CrossRef]
14. Alexe, S.; Blackstone, E.; Hammer, P.L.; Ishwaran, H.; Lauer, M.S.; Pothier Snader, C.E. Coronary risk prediction by logical analysis of data. *Ann. Oper. Res.* **2003**, *119*, 15–42. [CrossRef]
15. Hammer, P.L.; Kogan, A.; Lejeune, M.A. Modeling country risk ratings using partial orders. *Eur. J. Oper. Res.* **2006**, *175*, 836–859. [CrossRef]
16. Hammer, P.L.; Kogan, A.; Lejeune, M.A. A logical analysis of banks' financial strength ratings. *Expert Syst. Appl.* **2012**, *39*, 7808–7821. [CrossRef]
17. Rudin, C.; Shaposhnik, Y. Globally-Consistent Rule-Based Summary-Explanations for Machine Learning Models: Application to Credit-Risk Evaluation. Available online: https://ssrn.com/abstract=3395422 (accessed on 12 February 2022).
18. Bagchi, P.; Lejeune, M.A.; Alam, A. How supply competency affects FDI decisions: Some insights. *Int. J. Prod. Econ.* **2014**, *147 Pt B*, 239–251. [CrossRef]
19. Dupuis, C.; Gamache, M.; Page, J.-F. Logical analysis of data for estimating passenger show rates at Air Canada. *J. Air Transp. Manag.* **2012**, *18*, 78–81. [CrossRef]
20. Mortada, M.-A.; Yacout, S.; Lakis, A. Fault diagnosis in power transformers using multi-class logical analysis of data. *J. Intell. Manuf.* **2014**, *25*, 1429–1439. [CrossRef]
21. Das, T.K.; Adepu, S.; Zhou, J. Anomaly detection in Industrial Control Systems using Logical Analysis of Data. *Comput. Secur.* **2020**, *96*, 101935. [CrossRef]
22. Ragab, A.; El-Koujok, M.; Poulin, B.; Amazouz, M.; Yacout, S. Fault diagnosis in industrial chemical processes using interpretable patterns based on Logical Analysis of Data. *Expert Syst. Appl.* **2018**, *95*, 368–383. [CrossRef]
23. Jocelyn, S.; Ouali, M.-S.; Chinniah, Y. Estimation of probability of harm in safety of machinery using an investigation systemic approach and Logical Analysis of Data. *Saf. Sci.* **2018**, *105*, 32–45. [CrossRef]
24. Yuan, Y.; Zhuang, H. A genetic algorithm for generating fuzzy classification rules. *Fuzzy Sets Syst.* **1996**, *84*, 1–19. [CrossRef]
25. Hayashi, Y.; Imura, A. Fuzzy neural expert system with automated extraction of fuzzy if then rules from a trained neural network. In Proceedings of the First International Symposium on Uncertainty Modeling and Analysis, College Park, MD, USA, 3–5 December 1990; pp. 489–494.
26. Kosko, B. *Neural Networks and Fuzzy Systems*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1992.
27. Lin, C.-T.; Lee, C.S.G. Neural-network-based fuzzy logic control and decision system. *IEEE Trans. Comput*. **1991**, *12*, 1320–1336. [CrossRef]
28. Weber, R. Automatic knowledge acquisition for fuzzy control applications. In *International Symposium on Fuzzy Systems*; Kyushu Institute of Technology: Fukuoka, Japan, 1992; pp. 9–12.
29. Yuan, Y.; Shaw, M.J. Induction of fuzzy decision trees. *Fuzzy Sets Syst.* **1995**, *69*, 125–139. [CrossRef]
30. Nguyen, H.S. Approximate Boolean Reasoning: Foundations and Applications in Data Mining. In *Transactions on Rough Sets V. Lecture Notes in Computer Science*; Peters, J.F., Skowron, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4100.

31. Aamir, K.M.; Sarfraz, L.; Ramzan, M.; Bilal, M.; Shafi, J.; Attique, M. A Fuzzy Rule-Based System for Classification of Diabetes. *Sensors* **2021**, *21*, 8095. [CrossRef] [PubMed]
32. Hussain, S.; Kim, Y.-S.; Thakur, S.; Breslin, J.G. Optimization of Waiting Time for Electric Vehicles Using a Fuzzy Inference System. *IEEE Trans. Intell. Transp. Syst*. **2022**, 1–12. [CrossRef]
33. Hussain, S.; Ahmed, M.A.; Kim, Y.-C. Efficient Power Management Algorithm Based on Fuzzy Logic Inference for Electric Vehicles Parking Lot. *IEEE Access* **2019**, *7*, 65467–65485. [CrossRef]
34. Kromer, P.; Platos, J.; Snasel, V.; Abraham, A. Fuzzy classification by evolutionary algorithms. In Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, USA, 9–12 October 2011; pp. 313–318. [CrossRef]
35. Bonates, T.O.; Hammer, P.L.; Kogan, A. Maximum patterns in datasets. *Discrete Appl. Math*. **2008**, *156*, 846–861. [CrossRef]
36. Bshouty, N.H.; Eiron, N. Learning monotone DNF from a teacher that almost does not answer membership queries. *J. Mach. Learn. Res*. **2003**, *3*, 49–57.
37. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: Los Altos, CA, USA, 1993.
38. Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning about Data*; Kluwer Academic Publishers: Norwell, MA, USA, 1992.
39. Dong, J.L. Efficient mining of emerging patterns: Discovering trends and differences. In Proceedings of the Fifth ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 15–18 August 1999; ACM Press: New York, NY, USA, 1999; pp. 43–52.
40. Lavrac, N. Subgroup Discovery Techniques and Applications. In *Lecture Notes in Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3518, pp. 2–14.
41. Chikalov, I.; Lozin, V.; Lozina, I.; Moshkov, M.; Nguyen, H.S.; Skowron, A.; Zielosko, B. *Three Approaches to Data Analysis: Test Theory, Rough Sets and Logical Analysis of Data*; Springer: Berlin/Heidelberg, Germany, 2013. [CrossRef]
42. Yan, K.; Miao, D.; Guo, C.; Huang, C. Efficient feature selection for logical analysis of large-scale multi-class datasets. *J. Comb. Optim.* **2021**, *42*, 1–23. [CrossRef]
43. Bertolazzi, P.; Felici, G.; Festa, P.; Lancia, G. Logic classification and feature selection for biomedical data. *Comput. Math. Appl.* **2008**, *55*, 889–899. [CrossRef]
44. Kuzmich, R.; Stupina, A.; Korpacheva, L.; Ezhemanskaja, S.; Rouiga, I. The Modified Method of Logical Analysis Used for Solving Classification Problems. *Informatica* **2018**, *29*, 467–486. [CrossRef]
45. Letham, B.; Rudin, C.; McCormick, T.H.; Madigan, D. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *Ann. Appl. Stat.* **2015**, *9*, 1350–1371. [CrossRef]
46. Gasnikov, A. (Ed.) *Introduction to Mathematical Modeling of Traffic Flows*; MCCME: Moscow, Russia, 2013.
47. Too, J.; Abdullah, A.; Mohd, S.N.; Ali, N.; Tengku, Z.; Tengku, N.S. Featureless EMG Pattern Recognition Based on Convolutional Neural Network. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *14*, 1291–1297.
48. Masich, I.S.; Kazakovtsev, L.A.; Stupina, A.A. Optimization Models for Detection of Patterns in Data. In Proceedings of the School-Seminar on Optimization Problems and Their Applications (OPTA-SCL 2018), Omsk, Russia, 8–14 July 2018; pp. 264–275.
49. Kuzmich, R.I.; Masich, I.S.; Stupina, A.A.; Kazakovtsev, L.A. Algorithmic procedure for constructing the truncated basic set of characteristics in the method of logical analysis of data. In Proceedings of the 30th International Business Information Management Association Conference IBIMA 2017-Vision 2020: Sustainable Economic Development, Innovation Management, and Global Growth, Madrid, Spain, 8–9 November 2017; pp. 5592–5597.
50. Antamoshkin, A.N.; Masich, I.S.; Kuzmich, R.I. Heuristics and criteria for constructing logical patterns in data. In Proceedings of the International Scientific and Research Conference on Topical Issues in Aeronautics and Astronautics (Dedicated to the 55th Anniversary from the Foundation of SibSAU), Krasnoyarsk, Russia, 6–10 April 2015. [CrossRef]
51. Goh, C.-K.; Tan, K.C. *Evolutionary Multi-Objective Optimization in Uncertain Environments: Issues and Algorithms*; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
52. Noghin, V.D. *Reduction of the Pareto Set. An Axiomatic Approach*; Springer: Berlin/Heidelberg, Germany, 2018. [CrossRef]
53. Hammer, P.L.; Kogan, A.; Simeone, B.; Szedmak, S. Pareto-optimal patterns in logical analysis of data. *Discrete Appl. Math*. **2004**, *144*, 79–102. [CrossRef]
54. Masich, I.S.; Kazakovtsev, L.A. A Branch-and-Bound Algorithm for a Pseudo-Boolean Optimization Problem with Black-Box Functions. *Facta Univ. Ser. Math. Inform.* **2018**, *33*, 337–360. [CrossRef]
55. Deb, K.; Pratar, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE T. Evolut. Comput*. **2002**, *6*, 182–197. [CrossRef]
56. Whitley, D. The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, CA, USA, 4–7 June 1989; pp. 116–121.
57. Kazakovtsev, L.; Rozhnov, I.; Shkaberina, G. Increasing Population Variability in Parallel Genetic Algorithms with a Greedy Crossover for Large-Scale p-Median Problems. *IJAI* **2021**, *19*, 152–194.
58. Eremeev, A.V. Genetic Algorithm with Tournament Selection as a Local Search Method. *Discret. Anal. Oper. Res*. **2012**, *19*, 41–53. [CrossRef]
59. Veldhuizen, D.A.; Lamont, G.B. *Multi Objective Evolutionary Algorithm Research: A History and Analysis*; Tech. Rep.; Department of Electrical and Computer Engineering; Graduate School of Engineering; Air Force Institute of Technology; Wright-Patterson Air Force Base: Dayton, OH, USA, 1998.

60. Dai, C.; Wang, Y. A New Multiobjective Evolutionary Algorithm Based on Decomposition of the Objective Space for Multiobjective Optimization. *J. Appl. Math*. **2014**, *2014*, 906147. [CrossRef]
61. Jong, E.D.D.; Pollack, J.B. Multi-objective methods for tree size control. *Genet. Program. Evolv. Mach*. **2003**, *4*, 211–233. [CrossRef]
62. Liang, J.; Liu, Y.; Xue, Y. Preference-driven Pareto front exploitation for bloat control in genetic programming. *Appl. Soft Comput.* **2020**, *92*, 106254. [CrossRef]
63. Fogel, D. A parallel processing approach to a multiple travelling salesman problem using evolutionary programming. In Proceedings of the Fourth annual Symposium on Parallel Processing, Fullerton, CA, USA, 4–6 April 1990; pp. 318–326.
64. Fogel, D. An evolutionary approach to the travelling salesman problem. *Biol. Cybern*. **1988**, *60*, 139–144. [CrossRef]
65. Subasi, M.; Avila, J. *A New Approach to Select Significant Patterns in Logical Analysis of Data*; Rutcor Research Report; Rutgers University: New Brunswick, NJ, USA, 2012. [CrossRef]
66. UCI Machine Learning Repository: Breast Cancer Wisconsin (Original) Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original) (accessed on 13 February 2022).
67. UCI Machine Learning Repository: Myocardial Infarction Complications Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/Myocardial+infarction+complications (accessed on 13 February 2022).
68. Frank, E.; Hall, M.A.; Witten, I.H. *The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann: Burlington, MA, USA, 2016.
69. Dada, E.G.; Ngene, C.; Daramola, C.Y. Performance Comparison of Machine Learning Techniques for Breast Cancer Detection. *NJEAS* **2017**, *6*, 1–8.
70. Sarmento, R. *Breast Cancer Wisconsin (Original) Data Set (Analysis with Statsframe ULTRA)*; Technical Report; University of Wisconsin Hospitals: Madison, WI, USA, 2019. [CrossRef]
71. Vizza, P.; Curcio, A.; Tradigo, G.; Indolfi, C.; Veltri, P. A Framework for the Atrial Fibrillation Prediction in Electrophysiological Studies. *Comput. Methods Programs Biomed*. **2015**, *120*, 65–76. [CrossRef]
72. Bashar, S.K.; Ding, E.; Walkey, A.; Mcmanus, D.; Chon, K. Atrial Fibrillation Prediction from Critically Ill Sepsis Patients. *Biosensors* **2021**, *11*, 269. [CrossRef]

applied sciences

MDPI

Article

Integer Search Algorithm: A New Discrete Multi-Objective Algorithm for Pavement Maintenance Management Optimization

Abdulraaof Alqaili [1,*], Mohammed Qais [2,*] and Abdullah Al-Mansour [1]

[1] Civil Engineering Department, Faculty of Engineering, King Saud University, Riyadh 11421, Saudi Arabia; amansour@ksu.edu.sa

[2] Electrical Engineering Department, Faculty of Engineering, King Saud University, Riyadh 11421, Saudi Arabia

* Correspondence: aalqaili@ksu.edu.sa (A.A.); mqais@ksu.edu.sa (M.Q.)

Abstract: Optimization techniques keep road performance at a good level using a cost-effective maintenance strategy. Thus, the trade-off between cost and road performance is a multi-objective function. This paper offers a new multi-objective stochastic algorithm for discrete variables, which is called the integer search algorithm (ISA). This algorithm is applied to an optimal pavement maintenance management system (PMMS), where the variables are discrete. The PMMS optimization can be achieved by maximizing the condition of pavement with a minimum cost at specified constraints, so the PMMS is a constrained multi-objective problem. The ISA and genetic algorithm (GA) are applied to improve the performance condition rating (PCR) of the pavement in developing countries, where the annual budget is limited, so a minimum cost for three years' maintenance is scheduled. Study results revealed that the ISA produced an optimal solution for multi-function objectives better than the optimal solution of GA.

Keywords: genetic algorithm; integer search algorithm; pavement maintenance management

Citation: Alqaili, A.; Qais, M.; Al-Mansour, A. Integer Search Algorithm: A New Discrete Multi-Objective Algorithm for Pavement Maintenance Management Optimization. *Appl. Sci.* **2021**, *11*, 7170. https://doi.org/10.3390/app11157170

Academic Editor: Peng-Yeng Yin

Received: 23 June 2021
Accepted: 29 July 2021
Published: 3 August 2021

1. Introduction

Roadways are considered the arteries of economic and social activities at the national and local levels. Furthermore, they are one of the fundamental foundations of development countries. Pavement is an essential component of road infrastructure. Its surface plays an important role in providing a safe and comfortable environment for users. During the pavement design life, the pavement is exposed to increasing traffic volumes with heavy loads, environmental adverse impacts, and poor use, which lead to significant deterioration of the pavement. To make the pavement function better, regular maintenance should be done to repair the degradation of the pavement. Therefore, through regular maintenance, a sufficient budget will be spent to maintain the pavement at an adequate condition. Actually, limited budgets are the biggest difficulties and challenges facing the pavement maintenance. For this reason, the pavement maintenance activities should be managed within the available budget and resources [1]. A good pavement management program will preserve the condition of all road sections at an adequate high performance with minimum cost without any reverse effects on traffic operation, environment, or social activity [2]. Therefore, decision-makers need to have the methodologies for maintenance management to attain a sufficient level of service at a minimum cost [3]. In other words, optimizing the available budget allocation is needed for decision-makers to assist them in achieving the specified objectives of pavement performance under available budget and authority constraints [4].

Initially, the priority (rank) was used to determine pavement treatment plans. When the priority is applied, the maintenance activities (alternatives) are allocated using a ranking system based on parameters such as traffic volume, road class, quality index, etc. These parameters are determined based on road condition data for the current year. Therefore, the best maintenance plan cannot be assured by using a prioritization approach, and multi-year planning will also be problematic. In the last few decades, the optimization approach has been a priority to create the best possible solution for pavement maintenance strategy. In early maintenance optimization, the decision-making process was based on single-objective optimization. However, the decision-making process in pavement maintenance involves many objectives (e.g., cost, performance, etc.) that conflict with each other, and the solution resulting from single-objective optimization may be unacceptable to other objectives. Therefore, a reasonable solution to a multi-objective problem is to search for a set of solutions that satisfy the requirements of several objectives [5].

The programming methods, such as stochastic and deterministic, are utilized to satisfy the constraints related to the requirements of the road network for the proper maintenance plan. Stochastic techniques (e.g., Markov chain) are useful for the insufficient data of pavement conditions [6]. While, a set of deterministic techniques were used to resolve maintenance problems, such as dynamic programming [7], goal programming [8], quadratic programming framework [9], and nonlinear mathematical program [10].

Many methods are used for multi-objective optimizations, such as the weighted sum method, which is used widely. By this approach, several objective functions are summated with suitable weight for each objective function. The criteria are weighted, and the comparison of sorting orders is made for outcome reliability [11]. The priority process is divided into two stages: defining maintenance activities, and then analysis of trade-offs is executed to introduce many prioritized activities [12]. The multi-criteria are generated and weighted to specify the assets of prioritization [13].

By using the theory of multi-attribute utility, an axiomatizing mathematical process is suggested to quantify and analyze the alternatives, which include many competing results [14–16]. This method is a good trial to link the objectives of maintenance treatments on asphalt pavements and define the choices of decision-making. On the other hand, the analytic hierarchy process to estimate the weights of criteria set and alternatives are used [17–20]. In this regard, the other options are compared with related standards set by using pair-wise comparisons, then the criteria weight is determined. Thus, hierarchically, the alternatives and criteria are formed.

The genetic algorithm (GA) is an evolutionary computation technique, which is widely applied to solve different objectives of maintenance problems. The ability of a genetic algorithm in the optimization process is to deal with either simple issues (e.g., linear optimization) or the complex computational problems (e.g., multimodality) and support decision-making procedure with a reasonable solution in the maintenance plan. For this reason, GA has been used widely in maintenance optimization. GA technique provides a solution for allocation the available funds to achieve the objectives constrained of the resource of central and regional agencies [21]. Also, the GA is utilized to address the nature of combinatorial maintenance programing in the network-level of roads [22]. Besides, GA is applied to attain optimum maintenance plans at sufficient level for pavement sections [22]. For constrained optimization, the GA is used to optimize the maintenance of rural roads network by minimizing cost and maximizing the road performance [23–25]. An adaptive hybrid GA, which contains GA and local search, is applied for improving the effectiveness and efficiency of solution searching for the optimal maintenance plan [26].

Also, discrete particle swarm optimization (PSO) is applied to optimize a multi-objective problem for pavement maintenance optimization. In this method, random solutions are created, and optimal searching is made by updating populations [27,28]. Additionally, a parameter-free velocity term is introduced to the barebones algorithm, which can provide a feasible decision-making process [29].

Dominance-based rough set approach is mainly used in the selection of the best solution when a large number of the possible solutions has existed. In other words, this method enhances the decision-making process for optimizing various maintenance activities to accomplish predefined objectives and make optimal allocation of the available budget. In this method, the rough set theory is used, making it possible to analyze the contradiction. The optimization is achieved by this method through two stages. In the first one, a set of solutions from the Pareto optimal is created. In the second one, through the decision-making process, the best solutions in the created set are indicated [30,31]. The interactive multi-objective optimization-dominance rough set approach is also used to support decision-making interaction to determine the optimal set of maintenance activities [32].

However, most of the metaheuristic optimization algorithms are based on the real random numbers, which are not suitable for the discrete problems. This paper presented a new discrete algorithm called integer search algorithm (ISA) to optimize the PMMS problem. To continue in searching for methods to find the optimal solution supporting the decision making of pavement maintenance, this paper presents a new discrete stochastic algorithm called integer search algorithm (ISA) for the pavement management system. Also, the ISA and GA are applied to improve the pavement management for the road networks in a developing country with a limited annual budget. Furthermore, the convergence curves of ISA and GA during the optimization process are compared.

2. Pavement Maintenance Model

2.1. Pavement Condition Evaluation

The state of the pavement plays a significant role in the decision-making process for pavement repair. The pavement repair activities are dependent on the current condition of the pavement. As a result, determining the best maintenance strategy necessitates assessing the state of the pavement. In order to describe the current pavement condition, the proposed study used the pavement condition rating (PCR) with the grading system from 0 (Failed) to 4 (Excellent) as presented in Table 1.

Table 1. Pavement condition rating for IRI

PCR	IRI (m/km)	Condition Description
0	>10	Failed: out of service
1	(6,10]	Poor condition: foremost maintenance is required
2	[4,6]	Fair condition: medium maintenance is required
3	[2,4)	Good condition: less maintenance is required
4	<2	Excellent condition: maintenance is not required

2.2. Pavement Preservation Treatments

Pavement treatments are selected based on the purpose of the maintenance strategies of the networks. For retarding the pavement deterioration and reducing the distresses, this study interested on preventive maintenance treatments. Five maintenance activities were used in this study to implement the pavement maintenance called do nothing, crack seal, slurry seal, thin asphalt overlay, and thick asphalt overlay. These activities are assigned codes which are M-00, M-01, M-02, M-03, and M-04, respectively.

2.3. Performance Jumps

Application of pavement preservation treatments on the pavement surface leads to improve the pavement condition. The effectiveness of preservation treatments occurs in the performance improvement after the performing treatments. Therefore, the treatment effects should be determined to know the pavement condition after the treatments application by defining the performance jumps. This study used the performance jumps developed by Morcous and Lounis [24] as shown in Table 2.

Table 2. Influence of repair decisions on PCR

PCR		**PCR after Maintenance**			
		1	**2**	**3**	**4**
PCR before maintenance	0	M-01	M-02	M-03	M-04
	1	—-	M-01	M-02	M-03
	2	—-	—-	M-01	M-02
	3	—-	—-	—-	M-01

3. Optimization Methodology

The PMMS is a discrete problem, where the input variables are the maintenance activities, which are integer numbers between [0,4]. Also, the output variables are the PCR of pavement, which are integer numbers between [0,4]. Most of the metaheuristic optimization calculations are based on the real numbers, which are not reasonable for the PMMS issue. Thus, this paper displayed a modern discrete calculation called integer search algorithm (ISA) to optimize the PMMS problem.

3.1. Integer Search Algorithm

Most of the stochastic algorithms are based on the real numbers, which are used in all engineering applications. However, the variables of the PMMS problem are integer numbers, where these stochastic algorithms need adaptation to be suitable for discrete problems. Therefore, this paper has proposed a new discrete algorithm suitable for the discrete problem, which is called integer search algorithm (ISA). The ISA algorithm is proposed based on the random generation of integer numbers. The exploration and exploitation of the ISA algorithm are controlled using parameters a and A, as in (1) and (2). The ISA algorithm updates the integer variables *X*(*i*), as in (3). The main merits of the ISA algorithm are simplicity and speed.

$$a = round\left(b - b \times \frac{Iter}{Max\_iter}\right) \tag{1}$$

$$A = Integer\ Random\ \in [-1,1] \times a \tag{2}$$

$$X(i) = X_{best}(i) + A \tag{3}$$

where *b* is a constant ≥2, $X_{best}(i)$ is the best position of variable *X*(*i*), *Iter* is the iteration number, and *Max_iter* is the maximum number of iterations. the flowchart of the ISA algorithm is depicted in Figure 1.

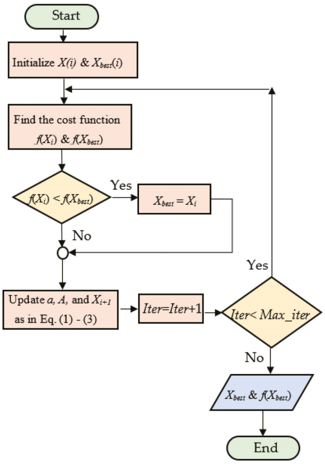

Start
Initialize $X(i)$ & $X_{best}(i)$
Find the cost function $f(X_i)$ & $f(X_{best})$
$f(X_i) < f(X_{best})$
Yes
$X_{best} = X_i$
No
Update a, A, and X_{i+1} as in Eq. (1) - (3)
$Iter = Iter+1$
$Iter < Max\_iter$
Yes
No
X_{best} & $f(X_{best})$
End

Figure 1. Flowchart of the integer search algorithm.

3.2. Objective Function

The optimization model formulation is performed in a way for keeping the level of the road performance at an acceptable level by using strategies of cost-effective maintenance. This constitutes using a multi-objective function in decision making to obtain the optimal maintenance plan.

As aforementioned, the PCR was used to evaluate the pavement condition in this study then one of the objective functions is maximizing the PCR value as shown in Equation (4). The other function is minimizing the cost to attain the optimal solution with minimum cost as possible as shown in Equation (5). In this study, the establishment of pavement maintenance strategies were constrained by the available budget for maintenance and the desired level of performance. The annual budget *Bt* constrains the objective function as in Equation (6). The Equation (7) is the constraint of the pavement performance. Only applying of one treatment is taken into account for each road section in each year as in Equation (8).

$$\boldsymbol{Max}\ \sum_{s=1}^{S}\sum_{j=1}^{m} PCR_{s,t}(X_{s,j}) \tag{4}$$

$$\boldsymbol{Min}\ \sum_{s=1}^{S}\sum_{j=1}^{m} C(X_{s,j})L_s W_S \tag{5}$$

Subjected to

$$\sum_{s=1}^{S} C\left(X_{s,j}\right) \times L_s \times W_s \leq\ B_t, \qquad \forall\, s = 1\ \ to\ S \tag{6}$$

$$\sum_{s=1}^{S} C\left(X_{s,j}\right) \times L_s \times W_s \leq B_t, \qquad \forall\, s = 1\ \ to\ S \tag{7}$$

$$\sum_{j=1}^{m} X_{s,j} = 1, \qquad \forall\, s = 1\ \ to\ S \tag{8}$$

where, $X_{s,j}$ maintenance application (option) for section s and treatment j, C is the cost of maintenance activity, L_s, and W_s are the length and width of the roadway section. The total maintenance cost of all sections is calculated as in (6).

3.3. Genetic Algorithm

The genetic algorithm (GA) is an evolutionary metaheuristic algorithm that has been applied to many engineering disciplines [33,34]. GAs work with a population of entities denoted by bit strings and change the population with random exploration and competition. In general, GAs consist of trials, for example, selection, crossover, and mutation. Generation is a procedure in which a novel cluster of entities is made by choosing the proper entities in the existing population. Crossover is the greatest influential worker in GAs. It constructs novel children by choosing two strings and exchange segments of their structures. The novel children may exchange the worst entities in the population. The mutation is a local worker with a very low likelihood, where its purpose is to adjust the value of a random position in a string.

In this work, the population of the GA algorithm is 100, where the population are initialized randomly. After that, this generated population is reproduced using crossover and mutation, where the crossover probability is set to 80% and the mutation probability is set to 0.05.

3.4. Optimization Procedure

The ISA algorithm is applied to find the optimal solution of the constrained multi-objective function shown in (4) and (5). The input variable is the maintenance activity

($X_{MA}(s, j)$) performed for each road section s and for treatment j. then the PCR of each section in a specific year is updated as in (9). The initialization of PCR, according to the constraints, is shown in the pseudo-code in Figure 2. The optimization procedure is applied for a number of sections and for a specified number of years, as shown in the pseudo-code in Figure 3.

$$PCR(s,t) = PCR(s,t-1) + X_{MA}(s,t) \tag{9}$$

```
Initialize the annual maintenance cost with large value
Do for number of years (t)
    While annual cost > annual budget
        generate integer number randomly ϵ [0,4] for maintenance activity (MA)
        Do for number of sections (dim)
            Update the PCR by Equation (9)
            While PCR > 4
                PCR = PCR-1
                $X_{MA}$ = $X_{MA}$-1
            End while
        End do
        Calculate the annual maintenance cost (C) by Equation (5)
    End while
End do
Output the best annual cost ($C_{best}$) = C, the $PCR_{best}$ = PCR, and $X_{MAbest}$ = $X_{MA}$
```

Figure 2. Pseudo-code of PCR initialization.

```
Initialize the best annual cost, best PCR, and best MA from Algorithm 1
While iterations < maximum number of iterations
    Calculate a by Equation (1)
    Do for number of years (t)
        Do for number of sections (dim)
            Calculate A by Equation (2)
            Calculate $X_{MA}$ by Equation (3)
            Update the PCR by Equation (9)
            While PCR > 4
                PCR = PCR-1
                $X_{MA}$ = $X_{MA}$-1
            End while
        End Do
        Calculate the annual maintenance cost (C) by Equation (5)
    End Do
    If the C less than $C_{a\text{-}best}$ and the sum of PCR less than $PCR_{best}$ then
        $C_{best}$ = C, the $PCR_{best}$ = PCR, and $X_{MAbes\ t}$= $X_{MA}$
    End if
End while
```

Figure 3. Pseudo-code of the application of the ISA algorithm for PMMS problem.

4. Case Study

The applicability of proposed algorithm was demonstrated by performing a case study. The data of the case study was obtained from the General Corporation of Roads and Bridges, Hajjah, Yemen, which is classified as a developing country. Roadway network contains 16 roads comprises arterial, main, and local access roads; which include 49 sections with 23 km total length. Code, length, width, and initial PCR for each road section are presented in Table 3. Annually, the maintenance activities are implemented within a specified available budget of around $80,000 based on the agency constraints. This study adopts a medium-term planning time horizon of three-year (2020–2022) to develop the pavement maintenance program. Maintenance costs related to repair decisions were collected from the General Corporation of Roads and Bridges in Yemen as 0.6, 1.08, 2.09, and 3 ($\$/\mathrm{m}^2$) for crack seal, slurry seal, thin asphalt overlay, and thick asphalt overlay, respectively.

Table 3. Road sections condition data

#	Street Name	Road Class	Section Code	Length (m)	Width (m)	Area (m^2)	IRI	Road Condition	PCR
1	Sana'a Road	Arterial	1R/100	92	7.1	656	5	Fair	2
2	Sana'a Road	Arterial	1L/100	243	7.1	1725	4	Fair	2
3	Sana'a Road	Arterial	2R/100	172	7.1	1221	6	Fair	2
4	Sana'a Road	Arterial	3/100	449	8	3592	7	Poor	1
5	Sana'a Road	Arterial	4/100	574	8	4592	12	Bad	0
6	Sana'a Road	Arterial	5/100	528	8	4224	11	Bad	0
7	Sana'a Road	Arterial	6/100	426	8	3408	13	Bad	0
8	Sana'a Road	Arterial	7/100	491	8	3928	8	Poor	1
9	Sana'a Road	Arterial	8/100	627	8	5016	9	Poor	1
10	Sana'a Road	Arterial	9/100	586	8	4688	13	Bad	0
11	Al-Hodaidah	Arterial	1R/200	211	5	1055	6	Fair	2
12	Al-Hodaidah	Arterial	1L/200	396	5	1980	3	Good	3
13	Al-Hodaidah	Arterial	2R/200	521	5	2605	5	Fair	2
14	Al-Hodaidah	Arterial	2L/200	875	5	4375	6	Fair	2
15	Al-Hodaidah	Arterial	3R/200	540	5	2700	5	Fair	2
16	Al-Hodaidah	Arterial	4/200	602	7	4214	3	Good	3
17	Al-Souq	Main	1/110	163	7	1141	7	Poor	1
18	Al-Souq	Main	2/110	238	7	1666	9	Poor	1
19	Al-Souq	Main	3/110	116	7	812	8	Poor	1
20	Al-Souq	Main	4/110	105	7	735	7	Poor	1
21	Al-Souq	Main	5/110	127	7	889	10	Poor	1
22	Al-Souq	Main	6/110	251	7	1757	6	Fair	2
23	Al-Dahreen	Main	1/120	285	7.2	2052	8	Poor	1
24	Al-Dahreen	Main	2/120	82	7.2	591	8	Poor	1
25	Al-Dahreen	Main	3/120	319	7.2	2297	6	Fair	2
26	Al-Dahreen	Main	4/120	276	7.2	1987	6	Fair	2
27	Al-Dahreen	Main	5/120	562	7.2	4046	9	Poor	1
28	Al-Dahreen	Main	6/120	107	7.2	770	9	Poor	1
29	Al-Dahreen	Main	7/120	586	7.2	4219	10	Poor	1
30	Al-Nasieryah	Local Access	1/116	694	8	5552	3	Good	3
31	Al-Qahirah	Local Access	1/112	329	6.5	2139	11	Bad	0
32	Al-Mohandis	Main	1/150	1348	6.3	8492	3	Good	3
33	Al-Mahjan	Main	1/130	567	7	3969	9	Poor	1
34	Al-Mahjan	Main	2/130	1028	7	7196	5	Fair	2
35	Qarn Habab	Main	1/160	156	6.1	952	13	Bad	0
36	Qarn Habab	Main	2/160	489	6.1	2983	11	Bad	0
37	Qarn Habab	Main	3/160	169	6.1	1031	12	Bad	0
38	Qarn Habab	Main	4/160	93	6.1	567	10	Poor	1
39	Na'aman	Local Access	1/113	725	6	4350	10	Poor	1
40	Al-Helah	Main	1/170	140	7.2	1008	7	Poor	1
41	Al-Helah	Main	2/170	307	7.2	2210	9	Poor	1
42	Al-Helah	Main	3/170	427	7.2	3074	8	Poor	1
43	Haorah	Local Access	1/111	761	8	6088	2	Good	3
44	Haorah	Local Access	2/111	352	8	2816	7	Poor	1
45	Al-Gharabi	Main	1/180	1401	7.1	9947	8	Poor	1
46	Al-Gharabi	Main	2/180	1926	7.1	13675	9	Poor	1
47	Al-Ma'enah	Main	1/140	889	7	6223	10	Poor	1
48	Dhola'ah	Local Access	1/114	315	6.1	1922	12	Bad	0
49	Qirwee	Local Access	1/115	431	4	1724	11	Bad	0

5. Results and Discussion

The ISA and GA algorithms are applied to maximize the total PCR of all road sections for three years and, at the same time, minimize the total cost of pavement maintenance. The PMMS model and ISA and GA algorithms are coded using MATLAB 2019b. The population size of both algorithms is set to 100, and the maximum number of iterations is set to 10,000. Therefore, the number of function evaluations is 100 × 10,000 = 1,000,000. The optimization process is executed using PC (Intel (R) Core (TM) i7–3770 CPU @ 3.40 GHz (8 CPUs), 16 GB, Windows 7–64 bits). Figure 4 shows that the attained cumulative PCR for 3 years and 49 sections using the ISA algorithm is 470, which is larger than that achieved by GA (PCR = 466). Furthermore, Figure 5 shows that the total cost of maximizing the PCR is minimized using ISA algorithm less than the total cost by using GA.

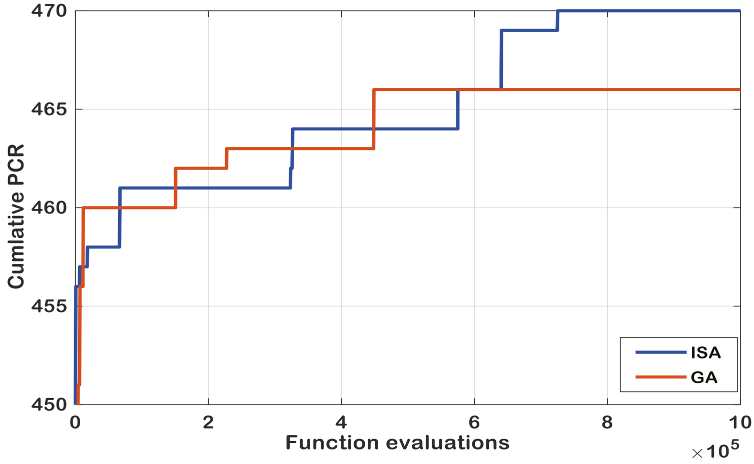


Figure 4. Convergence of the cumulative PCR for 3 years and 49 sections.

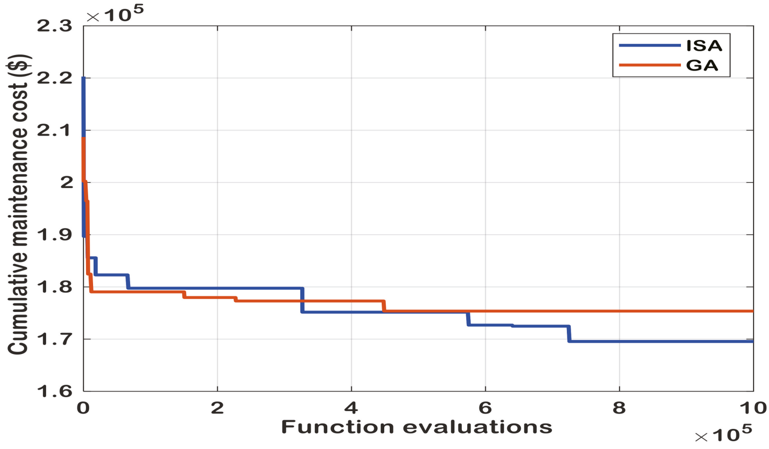


Figure 5. Convergence of the cumulative maintenance cost for 3 years and 49 sections.

Table 4 shows the optimal maintenance plan to attain acceptable performance within the available annual maintenance budget ($80,000). Therefore, the maintenance activities are distributed for all 49 sections for 3 years. For the first year t = 1 and for Sana'a road that contains 10 sections (from 1 to 10), sections 2, 6, 5, and 10 are not maintained M-00 (blue colored). However, sections 1, 4, 7, 8, and 9 are maintained by the maintenance activity M-01 (yellow colored), and Section 3 is maintained by using M-02 (green colored). In the second year, sections 1 and 3 are not maintained (M-00). However, sections 2, 5, 8, and 9 are maintained by using M-01, and sections 4, 6, 7, and 10 are maintained by using M-02.

Finally, in the third year, sections 2, 3, 4, and 7 are not maintained. However, sections 1, 5, 6, 8, 9, and 10 are maintained by using M-01. The last column in Table 4 shows that the PCR of about 36 sections improved to rank 4. However, 13 sections still need maintenance, which can be maintained in the third year because the total cost is low this year.

Table 4. Optimal maintenance activity plan by ISA algorithm to improve the pavement condition.

Section	Input Variables of ISA Algorithm			Section	Input Variables of ISA Algorithm		
s	$X_{MA}(s, t = 1)$	X_{MA} $(s, t = 2)$	X_{MA} $(s, t = 3)$	s	$X_{MA}(s, t = 1)$	X_{MA} $(s, t = 2)$	X_{MA} $(s, t = 3)$
1	1	0	1	26	2	0	0
2	0	1	0	27	1	2	0
3	2	0	0	28	3	0	0
4	1	2	0	29	1	0	1
5	0	1	1	30	0	1	0
6	0	2	1	31	2	2	0
7	1	2	0	32	0	0	0
8	1	1	1	33	0	1	1
9	1	1	1	34	0	0	0
10	0	2	1	35	4	0	0
11	2	0	0	36	2	1	0
12	1	0	0	37	2	1	1
13	2	0	0	38	3	0	0
14	0	2	0	39	3	0	0
15	2	0	0	40	3	0	0
16	0	1	0	41	2	1	0
17	3	0	0	42	0	3	0
18	2	1	0	43	0	1	0
19	3	0	0	44	0	3	0
20	0	3	0	45	0	0	0
21	3	0	0	46	0	0	0
22	2	0	0	47	1	0	0
23	2	0	1	48	2	2	0
24	2	1	0	49	4	0	0
25	2	0	0				

Figure 6 shows how the PCR is maximized by 193% from 1.22 in 2019 to 3.59 in 2022. The PCR maximization is constrained with the predefined annual budget ($80,000) and $240,000 for three years. Also, Figure 7 shows the annual maintenance cost for three years scheduled maintenance, where the average cost of three years is $56,515,446. Table 5 displays the obtained optimal total cost using the ISA algorithm is $169,546.34, where the cost-saving is about 29.4%. However, the total cost using GA is about $178,000, which is higher than the cost by using ISA.

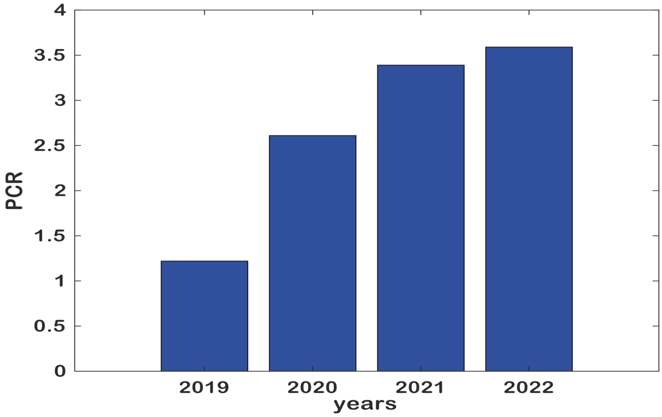


Figure 6. Improvement of PCR.

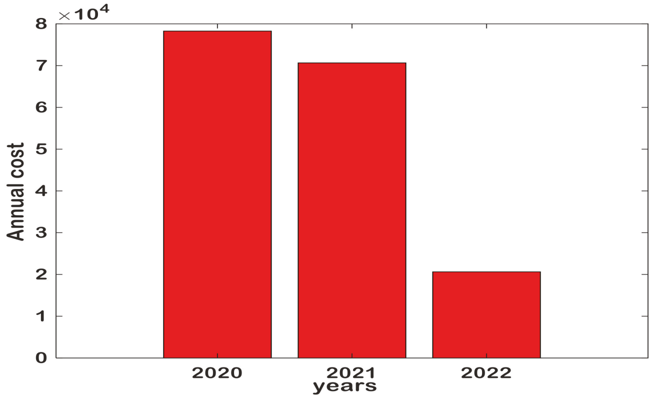


Figure 7. Annual maintenance cost during three years.

Table 5. Total annual PCR and its maintenance cost.

Year	*t* = 0	*t* = 1	*t* = 2	*t* = 3	Sum of *t* = 1, 2, 3
Total annual PCR	60	128	166	176	470
Average annual PCR	1.22	2.61	3.39	3.59	9.59
Annual cost ($)	–	78,272.06	70,650.90	20,623.38	169,546.34

Furthermore, the effectiveness of the results is statistically analyzed using the range and standard deviation of the PCR of the maintained roads in each year as depicted in Figure 8. Where the standard deviation and PCR range are decreasing with the planned annual maintenance. Moreover, the ISA algorithm utilized the maintenance activity M-01 about 21%, however, M-04 used rarely ~1% as depicted in Figure 9. These results reveal that the pavement maintenance of case study can be done effectively with the lowest cost. In addition, Figure 10 shows that the maintenance activities are optimally distributed between road classes. For the arterial roads, M-01 is the most used maintenance activity, however M-04 is never used. For local access roads, all maintenance activities are used, where M-02 is mostly used. For the main roads, all maintenance activities are applied, where the M-01 is mostly used. These results prove the effectiveness of the proposed optimization method for lowest cost of pavement maintenance management.

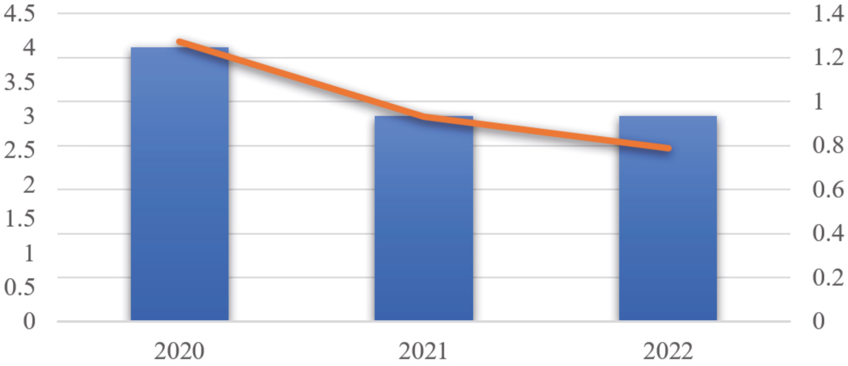


Figure 8. Range and standard deviation of PCR.

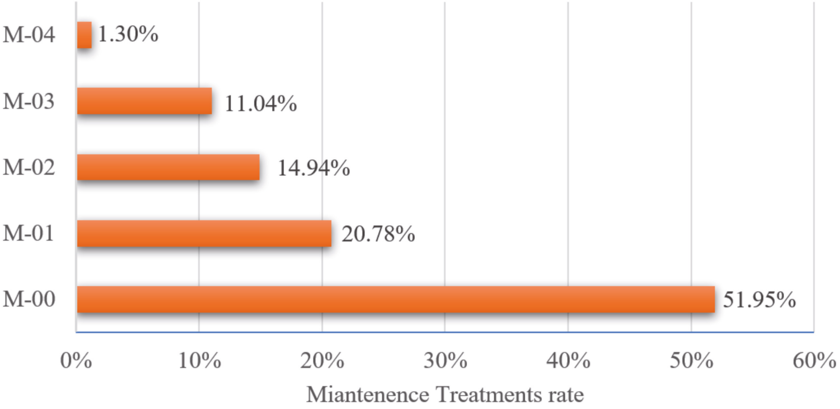


Figure 9. Optimal distribution of maintenance activities.

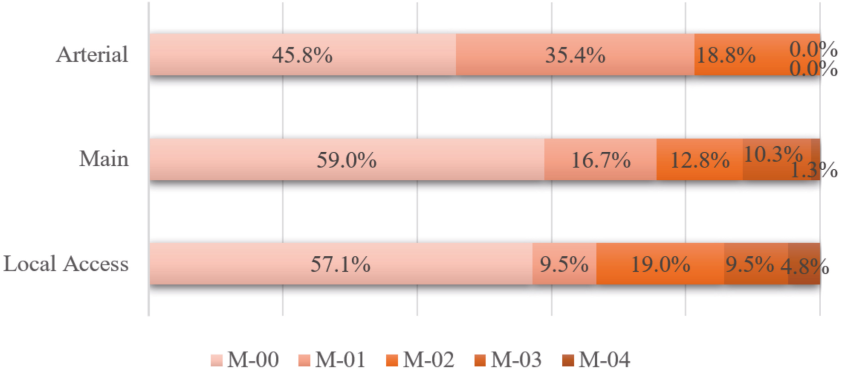


Figure 10. Optimal distribution of maintenance activities for road classes.

6. Conclusions

This paper has proposed a new discrete multi-objective stochastic algorithm based on the stochastic integer numbers, which is called integer search algorithm (ISA). Then, the ISA algorithm is applied to solve the constrained multi-objective function (pavement maintenance management (PMMS)). The main objectives of the PMMS problem are maximizing the performance of road pavements and minimizing the maintenance cost and constrained by the annual budget. In this paper, the case study is selected for Hajjah city in Yemen (developing country), where the number of roads is 16, and the number of sections is 49. Therefore, the number of input and output variables of PMMS is 49. The ISA and GA algorithms are applied to maintain all road sections in three years, so the maintenance activities are distributed during these years to obtain minimum cost. The optimization results revealed that the ISA achieved higher total PCR and lower total cost than that achieved by the GA algorithm. Also, the results show that the PCR is increased to 193% of the initial PCR with 29% cost-savings.

Author Contributions: Conceptualization, A.A.; Funding acquisition, A.A.-M.; Investigation, M.Q.; Methodology, A.A.; Resources, A.A.-M.; Software, M.Q.; Supervision, A.A.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Authors acknowledge the research center of college of engineering at King Saud University for the financial support to carry out this research paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. France-Mensah, J.; O'Brien, W.J. Developing a Sustainable Pavement Management Plan: Tradeoffs in Road Condition, User Costs, and Greenhouse Gas Emissions. *J. Manag. Eng.* **2019**, *35*, 04019005. [CrossRef]
2. Fwa, T.F.; Chan, W.T.; Hoque, K.Z. Multiobjective optimization for pavement maintenance programming. *J. Transp. Eng.* **2000**, *126*, 367–374. [CrossRef]
3. Wei, J.; Kong, Y.; Fu, Z. Multi-objective optimization for pavement maintenance and rehabilitation strategies. In Proceedings of the International Conference on Transportation Engineering 2009, Chengdu, China, 25–27 July 2009; pp. 2919–2924.
4. Torres-Machi, C.; Pellicer, E.; Yepes, V.; Chamorro, A. Towards a sustainable optimization of pavement maintenance programs under budgetary restrictions. *J. Clean. Prod.* **2017**, *148*, 90–102. [CrossRef]
5. Konak, A.; Coit, D.W.; Smith, A.E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 992–1007. [CrossRef]
6. Abaza, K.A. Iterative linear approach for nonlinear nonhomogenous stochastic pavement management models. *J. Transp. Eng.* **2006**, *132*, 244–256. [CrossRef]
7. Butt, A.A.; Shahin, M.Y.; Carpenter, S.H.; Carnahan, J.V. Application of Markov process to pavement management systems at network level. In Proceedings of the 3rd International Conference on Managing Pavements, San Antonio, TX, USA, 22–26 May 1994; Volume 2, pp. 159–172.
8. Wu, Z.; Flintsch, G.W.; Chowdhury, T. Hybrid multiobjective optimization model for regional pavement-preservation resource allocation. *Transp. Res. Rec.* **2008**, *2084*, 28–37. [CrossRef]
9. Durango-Cohen, P.L.; Sarutipand, P. Maintenance optimization for transportation systems with demand responsiveness. *Transp. Res. Part C Emerg. Technol.* **2009**, *17*, 337–348. [CrossRef]
10. Gu, W.; Ouyang, Y.; Madanat, S. Joint optimization of pavement maintenance and resurfacing planning. *Transp. Res. Part B Methodol.* **2012**, *46*, 511–519. [CrossRef]
11. Dissanayake, S.; John Lu, J.; Chu, X.; Turner, P. Use of multicriteria decision making to identify the critical highway safety needs of special population groups. *Transp. Res. Rec.* **1999**, *1693*, 13–17. [CrossRef]
12. Farhan, J.; Fwa, T.F. Incorporating priority preferences into pavement maintenance programming. *J. Transp. Eng.* **2012**, *138*, 714–722. [CrossRef]
13. Lambert, J.H.; Wu, Y.-J.; You, H.; Clarens, A.; Smith, B. Climate change influence on priority setting for transportation infrastructure assets. *J. Infrastruct. Syst.* **2013**, *19*, 36–46. [CrossRef]

14. Gharaibeh, N.G.; Chiu, Y.-C.; Gurian, P.L. Decision methodology for allocating funds across transportation infrastructure assets. *J. Infrastruct. Syst.* **2006**, *12*, 1–9. [CrossRef]
15. Giustozzi, F.; Crispino, M.; Flintsch, G. Multi-attribute life cycle assessment of preventive maintenance treatments on road pavements for achieving environmental sustainability. *Int. J. Life Cycle Assess.* **2012**, *17*, 409–419. [CrossRef]
16. Li, Z.; Sinha, K.C. Methodology for multicriteria decision making in highway asset management. *Transp. Res. Rec.* **2004**, *1885*, 79–87. [CrossRef]
17. Farhan, J.; Fwa, T.F. Pavement maintenance prioritization using analytic hierarchy process. *Transp. Res. Rec.* **2009**, *2093*, 12–24. [CrossRef]
18. Šelih, J.; Kne, A.; Srdić, A.; Žura, M. Multiple-criteria decision support system in highway infrastructure management. *Transport* **2008**, *23*, 299–305. [CrossRef]
19. Shelton, J.; Medina, M. Integrated multiple-criteria decision-making method to prioritize transportation projects. *Transp. Res. Rec.* **2010**, *2174*, 51–57. [CrossRef]
20. Sun, L.; Gu, W. Pavement condition assessment using fuzzy logic theory and analytic hierarchy process. *J. Transp. Eng.* **2011**, *137*, 648–655. [CrossRef]
21. Chan, W.T.; Fwa, T.F.; Tan, J.Y. Optimal fund-allocation analysis for multidistrict highway agencies. *J. Infrastruct. Syst.* **2003**, *9*, 167–175. [CrossRef]
22. Chootinan, P.; Chen, A.; Horrocks, M.R.; Bolling, D. A multi-year pavement maintenance program using a stochastic simulation-based genetic algorithm approach. *Transp. Res. Part A Policy Pract.* **2006**, *40*, 725–743. [CrossRef]
23. Mathew, B.S.; Isaac, K.P. Optimisation of maintenance strategy for rural road network using genetic algorithm. *Int. J. Pavement Eng.* **2014**, *15*, 352–360. [CrossRef]
24. Morcous, G.; Lounis, Z. Maintenance optimization of infrastructure networks using genetic algorithms. *Autom. Constr.* **2005**, *14*, 129–142. [CrossRef]
25. Pilson, C.; Hudson, W.R.; Anderson, V. Multiobjective optimization in pavement management by using genetic algorithms and efficient surfaces. *Transp. Res. Rec.* **1999**, *1655*, 42–48. [CrossRef]
26. Santos, J.; Ferreira, A.; Flintsch, G. An adaptive hybrid genetic algorithm for pavement management. *Int. J. Pavement Eng.* **2019**, *20*, 266–286. [CrossRef]
27. Shen, Y.; Bu, Y.; Yuan, M. A novel chaos particle swarm optimization (PSO) and its application in pavement maintance decision. In Proceedings of the 2009 4th IEEE Conference on Industrial Electronics and Applications, Xi'an, China, 25–27 May 2009; pp. 3521–3526.
28. Tayebi, N.R.; Moghadasnejhad, F.; Hassani, A. Analysis of pavement management activities programming by particle swarm optimization. In Proceedings of the International Conference on Advances in Electrical & Electronics, Trivandrum, Kerala, India, 1 January 2010; pp. 149–154.
29. Mahmood, M.; Mathavan, S.; Rahman, M. A parameter-free discrete particle swarm algorithm and its application to multi-objective pavement maintenance schemes. *Swarm Evol. Comput.* **2018**, *43*, 69–87. [CrossRef]
30. Greco, S.; Matarazzo, B.; Słowiński, R. Dominance-based rough set approach to interactive multiobjective optimization. In *Multiobjective Optimization*; Springer: London, UK, 2008; pp. 121–155.
31. Augeri, M.G.; Colombrita, R.; Greco, S.; Lo Certo, A.; Matarazzo, B.; Slowinski, R. Dominance-based rough set approach to budget allocation in highway maintenance activities. *J. Infrastruct. Syst.* **2011**, *17*, 75–85. [CrossRef]
32. Augeri, M.G.; Greco, S.; Nicolosi, V. Planning urban pavement maintenance by a new interactive multiobjective optimization approach. *Eur. Transp. Res. Rev.* **2019**, *11*, 17. [CrossRef]
33. Zhao, M.; Chen, Z.; Blaabjerg, F. Optimisation of electrical system for offshore wind farms via genetic algorithm. *IET Renew. Power Gener.* **2009**, *3*, 205–216. [CrossRef]
34. Shin, H.-U.; Lee, K.-B. Optimal design of a 1 kW switched reluctance generator for wind power systems using a genetic algorithm. *IET Electr. Power Appl.* **2016**, *10*, 807–817. [CrossRef]

applied sciences

MDPI

Article

Optimization of Carsharing Fleet Placement in Round-Trip Carsharing Service

Boonyarit Changaival [1,†], Kittichai Lavangnananda [2,*,†], Grégoire Danoy [1,†], Dzmitry Kliazovich [3,†], Frédéric Guinand [4,†], Matthias Brust [5,†], Jedrzej Musial [6,†] and Pascal Bouvry [1,†]

[1] Interdisciplinary Centre for Security, Reliability and Trust (SnT)-Faculty of Science, Technology and Medicine/Computer Science and Communications(FSTM/CSC), University of Luxembourg, Esch-sur-Alzette, 4364 Luxembourg, Luxembourg; b.c.boonyarit@gmail.com (B.C.); gregoire.danoy@uni.lu (G.D.); pascal.bouvry@uni.lu (P.B.)

[2] School of Information Technology (SIT), King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand

[3] ExaMotive S.A., Esch-sur-Alzette, 4263 Luxembourg, Luxembourg; kliazovich@ieee.org

[4] Laboratoire d'Informatique, du Traitement de l'Information et des Systéme (LITIS), Faculty of Science and Technology, Normandie University, 76600 Le Harve, France; frederic.guinand@univ-lehavre.fr

[5] Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Esch-sur-Alzette, 4364 Luxembourg, Luxembourg; matthias.brust@uni.lu

[6] Faculty of Computing and Telecommunications, Poznan University of Technology, 60-965 Poznan, Poland; jedrzej.musial@cs.put.poznan.pl

* Correspondence: kitt@sit.kmutt.ac.th

† These authors contributed equally to this work.

Citation: Changaival, B.; Lavangnananda, K.; Danoy, G.; Kliazovich, D.; Guinand, F.; Brust, M.R.; Musial, J.; Bouvry, P. Optimization of Carsharing Fleet Placement in Round-Trip Carsharing Service. *Appl. Sci.* **2021**, *11*, 11393. https://doi.org/10.3390/app112311393

Academic Editor: Juan-Carlos Cano

Received: 16 October 2021
Accepted: 24 November 2021
Published: 1 December 2021

Abstract: In a round-trip carsharing system, stations must be located in such a way that allow for maximum user coverage with the least walking distance as well as offer certain degrees of flexibility for returning. Therefore, a balance must be stricken between these factors. Providing a satisfactory system can be translated into an optimization problem and belongs to an NP-hard class. In this article, a novel optimization model for the round-trip carsharing fleet placement problem, called Fleet Placement Problem (FPP), is proposed. The optimization in this work is multiobjective and its NP-hard nature is proven. Three different optimization algorithms: PolySCIP (exact method), heuristics, and NSGA-II (metaheuristic) are investigated. This work adopts three real instances for the study, instead of their abstracts where they are most commonly used. They are two instance:, in the city of Luxembourg (smaller and larger) and a much larger instance in the city of Munich. Results from each algorithm are validated and compared with solution from human experts. Superiority of the proposed FPP model over the traditional methods is also demonstrated.

Keywords: carsharing system; fleet placement; metaheuristic algorithm; multiobjective optimization; NP-hard problem; NSGA-II; optimization; PolySCIP

1. Introduction

It is undeniable that efficient management of transportation has become one of the major problems in cities across the globe due to its impact on the environment and quality of life. Carsharing is one of many means of transportation nowadays and has received positive support from communities and governments. Its success can be seen in several countries, such as Germany, which has the biggest carsharing market in Europe with over 2 million registered users, 170 service providers, and over 16,000 vehicles available in 740 cities [1,2]. Coupling with the increasing awareness in environmental problems, the concept of green mobility is also promoted through the electrical carsharing service [3]. For instance, it has been highlighted that cars are used for transportation more than trains and planes in Germany and that carsharing positions itself is an intermediate mean to fill the gap between public transport and personal cars [4]. Another success was reported in the United Kingdom where the government provided support to extend the user base to 600,000 individuals by 2020 to reduce traffic and parking problems [2].

The carsharing model can be divided into free-floating and station-based [2]. Free-floating carsharing offers the highest degree of flexibility to the users. They pick up the nearby vehicle to start a trip and drop it off anywhere in the city to end the trip. However, flexibility comes with a high operational cost for the company, which needs to maintain a high density of vehicles even in low-demand areas of the city in order to cope with low levels of utilization. Additionally, vehicles that end up in low demand areas need to be reallocated. An example of well-known free-floating carsharing companies are SHARE NOW (which is the merge of Car2Go and DriveNow) [1].

In station-based carsharing, fleet vehicles are stationed in densely populated areas of the city and need to be returned to one of these locations after completing the trip (one-way) or to the same pick up area (round-trip). As a result, station-based services are less flexible, with an advantage of easier implementation and management. Station-based services require fewer cars, as carsharing operators can place vehicles in densely populated and high-demand zones of the city only, and no fleet relocation is typically required. An example for easier implementation and management is taken from bike-sharing. The bicycles need to be relocated everyday to maintain the service function, which is relatively easy because of their sizes. However, car relocation is more difficult and more expensive, which is an even bigger burden to carsharing operators. In addition, with the trend of electronic cars, a charging station is easier to implement in the station-based services due to having fewer stations to implement than the free-float services.

Therefore, a satisfactory solution associated with carsharing is multifarious. The work in this article is concerned with maximum user coverage and ease of access to the service (i.e., shortest distance to a station and flexibility in returning). It comprises the following:

1. Developing a concept of station and their locations that maximize user coverage while giving a certain degree of flexibility when returning a car;
2. Maintaining the right balance between user coverage and ease of access to the service;
3. Considering or designing a suitable metric which can be used to determine the ease of access for users at a global scale.

In this work, we propose a new method attempting to optimize the fleet placement in the station-based round-trip, will will be the first to tackle fleet allocation in round-trip carsharing. The model of this problem is called fleet placement problem (FPP). Fleet placement is really tedious and is usually performed manually by experts and hence is prone to errors due to lack of precision. The proposed methodology aims to maximize customer coverage, while minimizing the maximum walking distance between customers and the nearest vehicle. These two objectives are in conflict, thus, resulting in a bi-objective problem. Unlike previous solutions, the proposed model incorporates highly detailed street-level map data containing footprints of the buildings. The contributions proposed in this work are: mathematical formulation of the novel fleet placement problem (FPP) and its NP-hardness proof, correlation analysis of the two problem objectives, and the comparison of results between the manual placement and state-of-the-art heuristic and metaheuristic algorithms.

The remainder of this paper is organized as follows. Related work on fleet location for carsharing services and similar location problems are presented in Section 2. The formulation of the FPP problem and the methodology to solve it are detailed in Section 3. In Sections 5 and 6 results from executions on real city instances are presented and discussed. Finally, conclusions and perspectives are provided in Section 7.

2. Related Work

In this section, the state-of-the-art on fleet placement and location problems, shared fleet placement, and the optimization methods used to address are analyzed.

2.1. Fleet Placement and Location Problems

In the following, the similarities of the fleet location problem with two classical optimization problems, i.e., the maximal covering location problem (MCLP) and the facility location problem (FLP) are discussed. These two problems and the proposed fleet placement problem (FPP) can be reduced to the set covering problem as shown in the proof in Section 3.

Church and ReVelle proposed the maximal covering location problem (MCLP) in 1974 [5] for facility and emergency siting. The objective is to maximize the partial coverage with a number of facilities, where each facility has a fixed coverage distance. MCLP is shown to be NP-hard, which means it becomes intractable and cannot be solved in an acceptable time by exact methods when the size of an instance is large [6]. MCLP has been applied in many real-world problems. Seargeant used MCLP as a base model for placing healthcare facilities based on the demographic data in the regions [7]. Schmid et al. formulate their ambulance siting problem as an MCLP with the integration of patients' data and traces of taxis in Vienna to estimate the traveling time to reach the patient [8]. Another example in telecommunication is from Ghaffarinasab et al. who proposed a bi-level version of the hub interdiction problem (also another variant of MCLP) [9]. MCLP was also extended to its multi-objective. Xiao et al. proposed a MCLP with two objectives which were facility cost and proximity minimization [10]. Kim et al. solved another bi-objective version of the MCLP where the aim was to maximize primary and backup coverage (overlapping coverage for reliability) [11]. Malekpoor et al. formulated the problem of electrification in a disaster relief camp as finding a set of locations to reduce the project cost and increase the share of systems between sites [12].

In the facility location problem (FLP), the objective is to find locations to place facilities to supply stores, while minimizing the maximum cost (p-center) or the average cost (p-median) [13]. In this problem, one constraint is to have all the stores covered while one store can be covered by only one facility [14]. One of the many interesting applications of the FLP is shown in [15] where they utilized the spatial information and studied the difference between the optimal facilities locations and the current ones. Another application is in siting rescue boat locations. It was modelled as a multi-objective problem which considers not only the response time to the incidents, but also the operating cost and working hours [16]. FLP was also used in telecommunication to find the location of GSM antennas as shown in [17,18].

These two problems are highly related to our fleet placement problem (FPP). The similarity between FPP and MCLP is that they both try to maximize the partial coverage, with a constraint of fixed coverage distance and fixed number of facilities. Meanwhile, the FLP objective is to minimize the maximum operating cost, which is well aligned with FPP, with a second objective, which is to minimize the maximum walking distance. Therefore, FPP can be seen as a combination of these two problems.

2.2. Shared Fleet Placement

Shared fleet placement can be formulated into MCLP or FLP (especially in round-trip carsharing service). However, there are other factors to be considered. In previous works on MCLP and FLP, they already have a list of preferred locations. These possible sites are evaluated by considering convenience factors such as parking cost, the proximity to essential facilities, and accessing time as presented in [19,20]. The solution was then a combination of selected sites to maximize user coverage. In fact, they are very similar to the facility location problem. Kumar and Bierlaire evaluated potential stations by the distance between the station and other facilities such as hospitals and train stations. They also had access to historical data to make a decision on where to place the station, which is not available in most cases [20]. Another popular approach is to locate the fleet by user demands [21–23]. Boyacı et al. proposed an optimization model to maximize the user coverage based on the demand and predicted destinations [21]. On the contrary, Lage et al. studied a method to identify the potential of city districts in a station-based one-way trip scenario where the demands were estimated from the taxi trips and customer profiles in Sao Paulo, Brazil [22]. Lastly, Schwer and Timpf proposed an idea for locating the fleet in round-trip carsharing by combining both user demands and proximity to other mean of transportation and other facilities into a model and utilized the open source data available from the government [23].

There are also works which focus on electric carsharing fleets, reflecting the increasing awareness of environmental issues and benefiting from governmental support. The electric carsharing fleet is more complicated than its fossil-fuel counterpart considering the additional constraints for battery/electrical load management of the car. Çalik and Fortz proposed a model for a one-way electric carsharing service which considered previously mentioned factors [24]. Since charging is very important in electric carsharing service, Jiao et al. [25] formulated their model to consider a situation where the user changes the drop off station. The charging station location optimization was presented by Brandstätter et al. [26] where the authors based the location on the source and destination of trips in both simulation and Vienna. Another example was proposed by Yıldız et al. [27] which consider a more realistic case where demand was stochastic and capacitated charging stations. In addition, the shared fleet placement is also studied in the bikesharing community where station locations and bike stocking are highly important as well [28–32].

2.3. Existing Resolution Approaches

Several algorithms were proposed to solve the aforementioned problems, ranging from exact methods, to heuristic and metaheuristic algorithms. Exact methods guarantee optimality, however, once the size of an NP-hard problem is too large, such methods (e.g., branch-and-bound, exhaustive search) cannot find solutions in reasonable time. In contrast, heuristic algorithms (e.g., greedy algorithms) are problem specific methods that permit to obtain an approximate solution in reasonable time. Finally, metaheuristic algorithms (e.g., genetic algorithm or simulated annealing), are general purpose algorithms, which can lead to very satisfactory solutions. A true benefit is their acceptable execution time, which for middle to large size instances is several orders of magnitude smaller than for exact methods [33].

For generic location problems such as MCLP and FLP, exact methods are usually used [7,8,12,16,34–36]. It is important to note that the problem instances tackled in the reported articles were of limited size. In fact, Zarandi et al. [36] reported that IBM CPLEX [37] cannot handle a problem with a large size of input (e.g., a city). Hence, once the problem size is too large, heuristic and metaheuristic algorithms are usually employed. Church and ReVelle [5] first compared two variations of heuristic algorithms (which add one facility location one at a time) and a branch and bound algorithm. The next attempt in solving MCLP was using a Lagrangian heuristic algorithm, which is a combination of the Lagrangian Relaxation approach and a greedy method [38]. Heuristic algorithms are still being used nowadays as shown in [39] to solve the FLP problem. Lastly, several works reported on the efficiency of metaheuristics in solving FLP and MCLP. Tabu Search (TS), Simulated Annealing (SA), Variable Neighborhood Search (VNS), and Genetic Algorithms (GA) were also considered in solving MCLP [36,40–43]. Metaheuristics were not only used to solve single-objective versions of these problems but also multi-objective ones. Xiao et al. [10] employed a Multi-Objective Evolutionary Algorithm (MOEA) to solve the bi-objective MCLP, which focused on facility cost and proximity minimization using a specific encoding scheme and dedicated operators. Kim and Murray [11] solved the bi-objective reliability-focused MCLP where it aimed to maximize primary and backup locations coverage with a heuristic algorithm and a Multi-Objective Genetic Algorithm (MOGA). Karasakal and Silav [44] utilized the crowding distance function from the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [45] in the Strength Pareto Evolutionary Algorithm II (SPEA2) [46] and reported that the new algorithm outperformed the original NSGA-II and SPEA-II. Ranjbartezenji et al. [47] proposed their modified version of NSGA-II and used it to solve bi-objective MCLP.

In shared fleet placement, the most common approach is to model it as a single-objective problem (weighted sum) and to rely on exact solvers such as CPLEX or MATLAB [20,21,25,48,49]. Several path-based heuristic algorithms were proposed [24,26]. In fact, due to the problem complexity, heuristic and metaheuristic algorithms have been attracting more attention recently [50]. Another approach is to determine the fleet locations through agent-based simulation [51].

To date, metaheuristic algorithms have not been applied to shared fleet placement problem. Previous works mainly either consider small-size instances of the problem (at most 800 potential locations by curation) and apply exact methods, or propose heuristic algorithms. The location curation is normally conducted by field experts and is essential to facilitate the applications of exact methods and proposed heuristic algorithms, but it can be very time consuming and prone to error. Therefore, we aim to eliminate the curation process (to be fully automated) which in turn, leaves decision makers to consider over 100,000 locations (in cities like Munich). General exact methods and previously proposed heuristic algorithms are too computationally expensive to apply. Therefore, metaheuristic algorithms are a suitable candidate to solve such big instances of fleet placement problem efficiently.

3. Optimization Model

The focus of this article is on the round-trip carsharing service, which received relatively less attention in the research community [52]. Hence, a novel approach for round-trip carsharing fleet placement is proposed. There are three benefits that the approach offers. The first benefit is that automation of the fleet placement process removes the need for traditional manual allocation. The second benefit is the higher placement precision with the inclusion of a Geographic Information System (GIS). Finally, the third benefit is in proposing an approachable fleet management problem to the research community that may also be beneficial to similar applications.

The proposed approach emphasizes the user coverage and ease of access with a constraint of being applicable to the real-world scenario by utilizing the two components mentioned below. These two goals are of real concern in practice and are often expressed by experts in this area.

1. Utilizing graph theory to implement graph model representing a street network
2. Multiobjective Optimization model with two objectives, maximizing user coverage and minimizing global walking distance between cars and users

In this section, the graph instance used in this article is first defined for the FPP since the graph instance is closely related with the problem definition. Second, the Fleet Placement Problem (FPP) model is formulated. Finally, FPP is proven that it is an NP-hard in a strong sense.

3.1. Graph Instance Definition

The street map can be modeled as an undirected weighted graph to represent users' ability to walk on the streets in both directions where they need to pick up or leave the shared vehicle (see Figure 1). The street-level graph is modeled as follows:

$$G = (V, E, P, W).$$

The set of nodes V is composed of two subsets: $V = S \cup B$, where S is the subset of street nodes (i.e., nodes on roads and streets), while subset B contains buildings. Both S and B are of type node, hence they are naturally members of vertices in G.

Buildings contain users. A weight $p_i \in P$ associated with each node $b_i \in B$ corresponds to the estimation of the number of people living in this building. The weight for each node in S is set to zero, making the assumption that target users are only located in buildings. This is similarly to the study taken by Daniels and Mulley [53]).

The set of edges E consists of two subsets: $E = R \cup L$, where R is a set of streets/roads, and L is a set of links connecting residential buildings to nearby streets.

Each edge $(u, v) \in R$ is valuated by a weight, $w_i \in W$, representing the walking distance from u to v where $u, v \in S$.

Each edge $(u, b) \in L$ where $u \in S$ and $b \in B$ is valuated by 0. In other words we consider the distance between the building and its nearest adjacent street is considered negligible. This process is called "snapping" and is common in every routing service where the starting point is first projected on a road before starting to build a route [54].

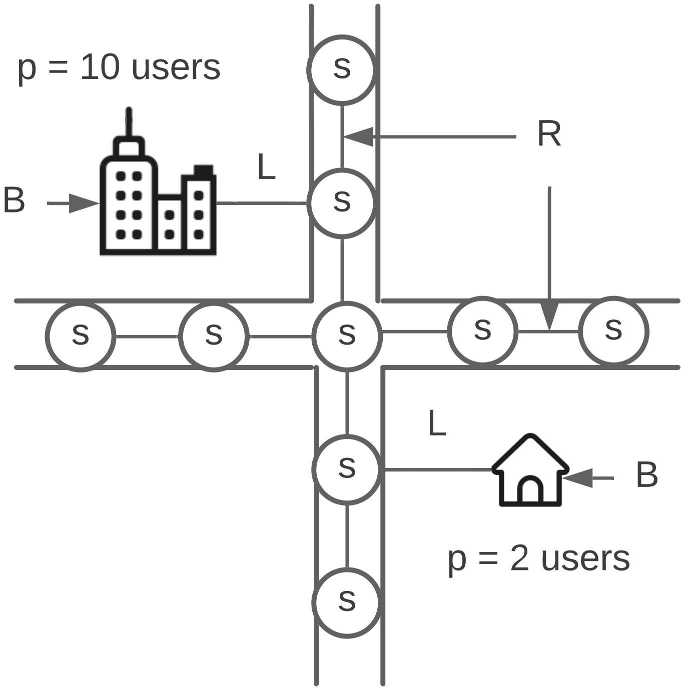


Figure 1. An illustration of a graph instance.

3.2. Fleet Placement Problem

The concept of a virtual station is created in this work, In the FPP model, carsharing stations are placed on the streets. A station is a virtual area on the city map defined by two elements, a center point, and a radius. Figure 2 illustrates this concept. A car is placed on a road which depicts a center point. A circle represents a radius of that center point. These two elements constitute the virtual station. A car can be picked up and returned to anywhere in that circle (station area). The coverage of a station is determined by the given maximum walking distance illustrated as a green line in the figure.

Due to the round-trip nature of the service, each car taken from the station needs to be returned to this station after completing the trip. The typical customers are people in residential areas (as shown in Figure 2) covered by the carsharing stations, who walk to pick up the nearest vehicle. Under the aforementioned assumptions and constraints, FPP can then be formulated.

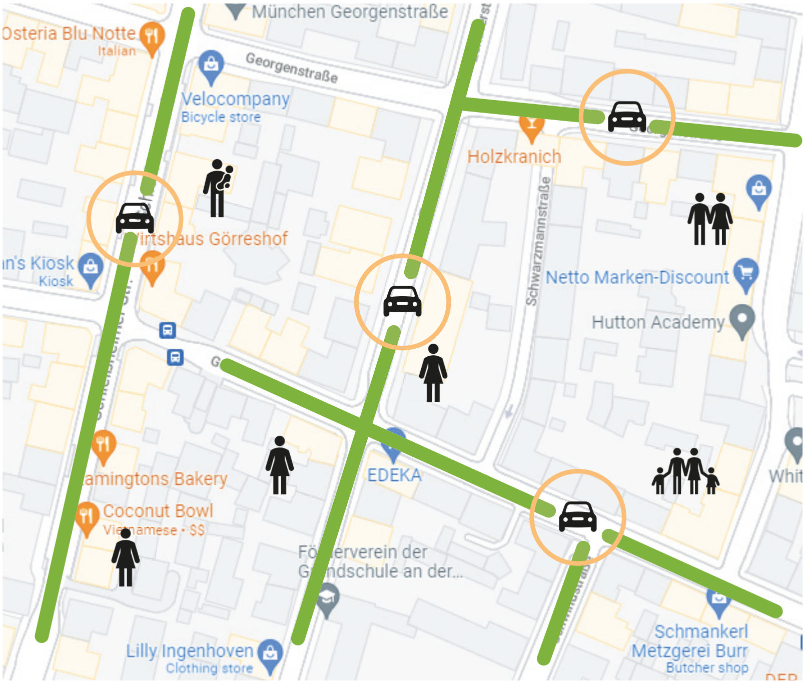


Figure 2. An illustration of a virtual area of a station on streets.

3.2.1. FPP Parameters

In the fleet placement problem (FPP), there are two types of parameters, inputs and instance parameters. Input parameters are the number of fleet stations, maximum walking distance, and station radius as shown in Table 1. Instance parameters are related to the problem graph instance, which are the set of street nodes, buildings, population, and distance between street nodes and buildings as summarized in Table 2. Stations are put on the street node in $S = \{s_1, s_2, \ldots, s_n\}$ and have a unique size r (see Figure 2). Users are assumed to start the trip from their residence (referred to as building) in B = {b_1, b_2, . . . , b_m} where each residence or building has a different number of users defined as $P = \{p_1, p_2, \ldots, p_m\}$. Each street node covers a set of buildings, which depends on the defined walking distance w and a set of distance $D = \{d_{11}, d_{12}, \ldots, d_{ij}\}$ where d_{ij} denotes walking distance between node s_i and building b_j where a distance function $d(s_i, b_j) = d_{ij}$. A station s_i covers a building b_j *iff* that station locates at most $w - r$ meters from the building.

Table 1. FPP input parameters.

Input Parameters	Description	Type
f	Maximum number of desired fleet stations.	$\mathbb{Z}^+$
w	Maximum walking distance allowed.	$\mathbb{R}^+_*$
r	Station area radius.	$\mathbb{R}^+_*, r \leq \frac{w}{2}$

Table 2. FPP instance parameters.

Instance Parameters	Description	Type
n	Number of street nodes.	$\mathbb{Z}^+$
m	Number of buildings.	$\mathbb{Z}^+$
i	Index for street nodes	$\mathbb{Z}^+$
j	Index for buildings	$\mathbb{Z}^+$
S	Set of street nodes (potential stations)	$S = \{s_1, s_2, \ldots, s_n\}$
B	Set of buildings (housing users)	$B = \{b_1, b_2, \ldots, b_m\}$
P	Set of population of buildings	$P = \{p_1, p_2, \ldots, p_m\}$.
D	Set of walking distances between street nodes and buildings.	$D = \{d_{11}, d_{12}, \ldots, d_{ij} : d_{ij} \in \mathbb{R}^+\}$

3.2.2. FPP Variables

The variable of FPP consists of three variables. The first variable is the set of the state of buildings C where c_ij represents the fact that building b_j is or is not covered by a station s_i. If the building b_j is covered then $c_{ij} = 1$ and $c_{ij} = 0$ otherwise. A building is covered when the center point of a station (at least) is located within walking distance smaller than $w - r$. The second variable is a set of state street nodes (S'), where s'_i represents the state of node s_i ($s_i \in S$). A node is active if it is the center point of a station. If s_i is active then $s'_i = 1$ and $s'_i = 0$ if s_i is inactive. It is important to note that a station can be active even if it does not cover any buildings. In this version, the station is able to accommodate only one vehicle at a time. Finally, the maximum global walking distance (z) denotes maximum walking distance from every selected stations to their covered buildings (hence, covered population) as shown in Table 3.

Table 3. FPP variables.

Decision Variables	Description	Type
C	Set of state of buildings	$C = \{c_{11}, c_{12}, \ldots, c_{nm}\}, c_{ij} \in \{0,1\}$
S'	Set of state of street nodes	$S' = \{s'_1, s'_2, \ldots, s'_m\}, s'_i \in \{0,1\}$
z	maximum global walking distance	$\mathbb{R}^+$

3.2.3. FPP Objectives

In the fleet placement problem, the objectives are to maximise the users coverage of a station and to minimise the maximum global walking distance between users and fleet stations. With these two objectives, the optimization model is formulated as follows:

$$
\begin{array}{lllr}
\max & \sum_{i=1}^{n}\sum_{j=1}^{m}(c_{ij} \times p_j) & & \\
\min & z = max_{i,j}(s'_i \times c_{ij} \times d_{ij}) & \forall i,j & \\
\text{s.t.} & \sum_{i=1}^{n} s'_i \leq f & & \text{(C1)} \\
 & (w-r)c_{ij} \geq s'_i(w-r) - (d_{ij} \times s'_i) & \forall i,j & \text{(C2)} \\
 & s'_i \geq c_{ij} & \forall i,j & \text{(C3)} \\
 & \sum_{i=1}^{n} c_{ij} = 1 & \forall j & \text{(C4)} \\
 & z \geq s'_i \times c_{ij} \times d_{ij} & \forall i,j & \text{(C5)} \\
 & z \geq 0 & & \text{(C6)} \\
 & c_{ij} \in \{0,1\} & \forall i,j & \text{(C7)} \\
 & s'_i \in \{0,1\} & \forall i & \text{(C8)} \\
 & i \in \{1,\ldots,n\} & & \text{(C9)} \\
 & j \in \{1,\ldots,m\} & & \text{(C10)}
\end{array}
$$

C denotes a constraint. Constraint 1 denotes that the number of stations cannot exceed the provided number f of fleet stations. Constraint 2, 3, and 4 restrict the model to consider any building b_j to be covered by one station and to be calculated only once for user coverage. The building b_j is covered *iff* there is at least one active station in its proximity (the proximity is defined by $w - r$). Constraint 5 finds the maximum walking distance of the active station s_i from all buildings b_j that it covers. Constraint 6 impose the global walking distance to always be positive. Constraints 7 and 8 indicate that there are only two states for street nodes (active: 0, inactive: 1) and buildings (not covered: 0, covered: 1). Finally, constraints 9 and 10 denote the domains of indices i and j accordingly.

3.3. NP-Hardness Proof

According to Garey and Johnson [55], any decision problem that can be reduced from an NP-complete problem, whether it is a member of NP or not, is not solvable in polynomial time unless P = NP since it is as hard as the NP-complete problem. In order to prove the NP-hardness of FPP, its computational complexity is analyzed. Therefore, the decision counterpart of the fleet placement problem (FPP)–FPP–D is introduced. The decision counterpart FPP–D inherits all parameters from FPP.

In this section, the NP-hardness of FPP, through proving the NP-completeness of FPP–D, is demonstrated. For FPP–D, the question is to determine whether there exists a solution with f station(s) such that all buildings are covered.

Proposition 1. *The FPP is NP-hard in the strong sense even if there is only one user in each building.*

Proof. We introduce a polynomial-time transformation to the FPP–D from the strongly NP-complete problem "Set Cover Problem (Minimum Cover Problem)" [55,56].

Set Cover Problem or SCP can be defined as follows: given a universe U of R elements, a collection of subsets of U, $G = \{g_1, g_2, g_3, ..., g_L\}$ and a positive integer $K \leq |G|$, the question is "Does G contain a cover for U of size K or less, i.e., a subset $G' \subset G$ with $|G'| \leq K$ such that every element of U belongs to at least one member of G'?"

Given an instance of SCP, we introduce the following instance of FPP–D. Firstly, let all buildings in B be the equivalence of universe U in SCP and $|B| = R$. Then, let S be the direct transformation of collection G where $S = \{s_1, s_2, s_3, ..., s_L\}$, such that $s_l = g_l, l = 1, 2, 3, \ldots, L$. In addition, we let $w, z = 1$ and $r = 0$ so that the building is covered if it is connected to the street node (s_l). With the prior assumption, the distances in matrix D are assumed to be one if the street node is a 1-hop neighbour of the building and zero, otherwise. Therefore matrix D reflects the membership of S and is used to constitute the membership of collection S in FPP–D. We also assume that there are L stations, hence $n = L$. Next, we assume $p_j = 1; j \in \{1, 2, 3, \ldots, |B|\}$ which means there is only user in building j. Let c_j be one if a collection s_i contains a building b_j where $i \in \{1, 2, 3, \ldots, n\}$ and $j \in \{1, 2, 3, \ldots, |B|\}$. With the aforementioned assumptions, $\sum_{j=1}^{R}(c_j \times p_j) = |B|$. Finally, we let the threshold value $f = K$.

Let X be a solution to SCP. A solution for FPP–D is constructed in which the buildings in B (U) are covered by f stations where $s'_l = g_l \in X$, such that $x_l = s'_l$, if $s'_l \in X$ and $x_l = \emptyset$ if $s'_l \notin X$. Since X is a cover of U (in SCP), all buildings in B are covered and the number of stations in the corresponding solution (for FPP–D) is $f = |X|$.

Now assume that there exists a solution Y in FPP–D with $|Y| \leq K$ and $|Y|$ should not exceed K, otherwise, $|Y| > K$ and the condition will not hold. Therefore, there are at most K station(s) with $y_i \neq \emptyset$. Since all buildings form B and all buildings in B belong to at least one member of Y, the selected stations with $y_i \neq \emptyset$ represents a solution to SCP, given a polynomial transformation from SCP to FPP-D. Since all input numbers in the FPP–D instance have a size most polynomial in the size of the input, FPP is strongly NP-hard. □

SCP (as an optimization problem) was proved to be polynomially non-approximable within the ratio $c \cdot \ln |G|$, for some constant $c > 0$ [57]. Therefore, we propose the following statement.

Statement 1. There exists no polynomial $(c \cdot \ln n)$- approximation algorithm for the FPP where n is the input size, unless $P = NP$.

4. Optimization Methods

In this section, the state-of-the-art algorithms used in our experiments, PolySCIP [58], heuristics [5], and the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [45] are described. Each of them represents a different category of problem solver.

4.1. PolySCIP

The strength of exact algorithms is the guarantee of reaching the global optimum, but the related computational cost can prevent their usage for large size NP-hard problems. Examples of classical exact algorithms are branch and bound, branch and cut, or A*. There exist also commercial exact solvers such as IBM CPLEX [37] and AMPL [59] but to our knowledge, these algorithms and solvers are only able to solve single objective optimization problems. This limitation led to research for multi-objective exact solvers and in 2016, PolySCIP was proposed [58].

PolySCIP employs a "Lifted Weight Space Approach" [58]. This approach first optimizes the objectives lexicographically. The weighted (single objective) optimization problem from the first phase is optimized by using positive weight vectors. This guides the algorithm in exploring the Pareto front in the problem space. If the new non-dominated solution is found, the old solution (the one that has been dominated) is discarded and the process continues until all non-dominated solutions are found. As a result, the outcome is a Pareto front instead of just one solution. The method was proven mathematically in finding all global optima by the authors.

4.2. Heuristic Algorithms

Heuristic algorithms can be simply described as a set of rules to follow. A rule can be as simple as taking whatever that is the best in that particular instance (see Algorithm 1). Numbers of variations of heuristic algorithms were used in solving both MCLP and carsharing fleet management. Church and ReVelle [5] first proposed a heuristic algorithm, which adds one facility location at a time. The latter has also been applied to solve shared fleet placement problems [20,21] and is used in the comparative study in this work as well.

Algorithm 1: Greedy search algorithm

```
Data: Number of locations ($N$), Potential locations($L$)
Result: List of selected locations($S$)
S ← ∅
for $l \in L$ do
    evaluate location $l$ using a fitness function
end
sort locations according to fitness score
$S \leftarrow N$ best fitness location ($l$)
Return $S$
```

This heuristic algorithm relies on an iterative search. The algorithm starts with an empty list of locations. Then, in each iteration, each location is evaluated according to the fitness function. The algorithm then adds the location with the highest fitness score in the list and that selected location is removed from the location pool, in order not to be re-selected in the subsequent iteration. This algorithm has a complexity of $O(sn)$ where s is the number of (desired) stations and n is the number of street nodes. The pseudocode of the algorithm is shown in Algorithm 2.

Algorithm 2: Iterative search algorithm

```
Data: Number of locations ($N$), Potential locations($L$)
Result: List of selected locations($S$)
S $\leftarrow \emptyset$
for $n \in \{1 \ldots N\}$ do
    for location(l) in L do
        evaluate location $l$ using a fitness function
    end
    sort location according to fitness score
    $S \leftarrow$ best fitness location
    remove facilities that are covered by location $l$
end
Return $S$
```

Even though the algorithm was first introduced to solve single objective optimization problems, there are several ways to adapt it to solve multi-objective problems, e.g., weighted sum, and ϵ-constraint. These variations can be extended further to yield an approximated front as a result. Using a weighted sum approach, each objective fitness is normalized as shown in Equation (1).

$$F_n = \frac{F_a - LB}{UB - LB}, \tag{1}$$

where F_n is the normalised fitness and F_a is the actual fitness (e.g., coverage, walking distance, or bi-objective) before normalization. UB is the upper bound (the highest fitness) and LB is the lower bound (the lowest fitness). With this weighted sum approach, the priority of each objective can be adjusted as shown in Equation (2). The iterative search can be launched multiple times with different weight ratios for each optimization objective. Let us for instance assume that there is the following list of weights: [(0.1,0.8,0.1), (0.33, 0.33, 0.33), (0.1, 0.1, 0.8)]. With these weights, up to three solutions can be reached.

$$F_t = \sum_{i=1}^{m} w_i F_i \tag{2}$$

where F_t is the total fitness score from the weighted sum and w_i is the weight associated to the objective i. Finally, F_i is the normalized fitness score of objective i. The granularity of weights can also be adjusted at the cost of a higher computation cost since more combinations of weights require more executions of the algorithm. Once a certain amount of solutions (decided by the decision maker) is collected, an approximated front can be constructed.

In this work, we study six variants of heuristic algorithms based on the greedy and iterative versions. They are;

1. Coverage-focused greedy algorithm;
2. Distance-focused greedy algorithm;
3. Bi-objective-focused greedy algorithm;
4. Coverage-focused iterative algorithm;
5. Distance-focused iterative algorithm;
6. Bi-objective-focused iterative algorithm.

The purpose is to establish a baseline for comparison and to evaluate the performance of the state-of-the-art heuristic algorithms against other algorithms.

4.3. Non-Dominated Sorting Genetic Algorithm-II (NSGA-II)

In the Evolutionary Algorithm (EA) approach to solving a problem, a well known concept called 'metaheuristic' can be concisely defined as a higher-level procedure or strategy for a partial search. Hence, a global optimum is not guaranteed, but it generally yields acceptable results. Metaheuristics usually contain a stochastic process, which make them non-deterministic. NSGA-II is a metaheuristic optimization algorithm that is based on Pareto-dominance [45]. Pareto-dominance is defined as follows:

$$\begin{aligned} z \succ z' \Leftrightarrow & \forall i \in \{1, 2, \ldots, n\}, z_i \leq z'_i \ \cup \\ & \exists j \in \{1, 2, \ldots, n\}, z_j < z'_j \ , \end{aligned}$$

where z and z' are vectors of objectives in Z and $z \succ z'$ means z dominates z'. If the selected solutions are both non-dominated, one of the parent solutions is selected at uniformly random. Other metaheuristic algorithms in this category are Simple Evolution Algorithm for multi-objective Optimization (SEAMO) [60], Strength Pareto Evolutionary Algorithm II (SPEA2) [46], and Pareto Envelope-based Selection Algorithm (PESA) [61]. NSGA-II was shown to be more efficient than the previously mentioned algorithms in a GSM antenna location problem (a variant of FLP) [17].

Among these algorithms, NSGA-II is renowned due to its numerous proven applications. It is largely based on the Genetic Algorithm (GA), starting from population initialization, selection of parents, crossover, and mutation to obtain a new population of solutions. Individuals in both the parent and offspring populations are sorted according to their rank, and the best solutions are chosen to create a new population. If individuals have the same rank, a density estimation based on the crowding distance to the surrounding individuals of the same rank is used. A new reference-point-based NSGA-II called NSGA-III is proposed, with the intention on solving problems with three or more objectives [62]. Hence, in this work, NSGA-II is selected as the problem is bi-objective. The pseudocode of NSGA-II is shown in Algorithm 3.

Algorithm 3: Nondominated Sorting Genetic Algorithm (NSGA-II)

```
population $\leftarrow$ InitializePopulation (size)
Evaluate (population)
NondominatedSort (population)
CrowdingDistance (population)
while termination criteria are not yet satisfied do
    parents $\leftarrow$ TournamentDCD (population)
    offspring $\leftarrow$ recombination+mutation(parents)
    Evaluate (offspring)
    NondominatedSort (population + offspring)
    CrowdingDistance (population + offspring)
    population $\leftarrow$ CrowdedComparison (population + offspring)
end
Return population
```

Next, the solution encoding population initialization and evolutionary operators employed in NSGA-II to solve the fleet placement problem are presented in detail. More details on NSGA-II's specific operators can be found in [45].

Solution Encoding: A solution is a string of fleet locations denoted by 'id numbers' from Openstreetmap. An example is shown in Figure 3, blue dots are candidate fleet locations on the streets. The shown numbers are ids from Openstreetmap. An example solution contains five locations (location number 1 to 5). Naturally, it is one of many possibilities in this example area. This encoding is more suitable than the binary encoding due to the size of the problem instances, which can be exceedingly large (more than 100,000 street nodes is possible in reality). The encoding contains no order and swapping genes in the chromosome does not change the fitness of the solution.

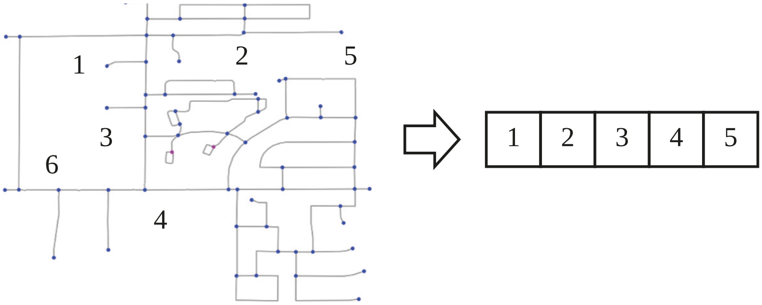


Figure 3. Example of a chromosome representing a possible solution. Street nodes are represented by IDs (numbers) and a solution is consisted of these IDs.

Population Initialization: A solution is initialized by randomly choosing (based on a uniform distribution law) street nodes from all street nodes in a problem instance. Note that each street node in the algorithm may at most be selected once in each solution during the initialization.

Crossover: A two-point crossover is adopted in this work. The process randomly selects two points in both solutions as starting and ending points for exchanging portions and recombines these portions to create two new solutions as shown in Figure 4. Although, this crossover process may introduce solutions with redundant placements, due to fitness score calculation (e.g., redundant locations lead to lower coverage) those solutions will be deemed as low quality and hence be eliminated in the next generation of selection. This reduces the execution time of the algorithm.

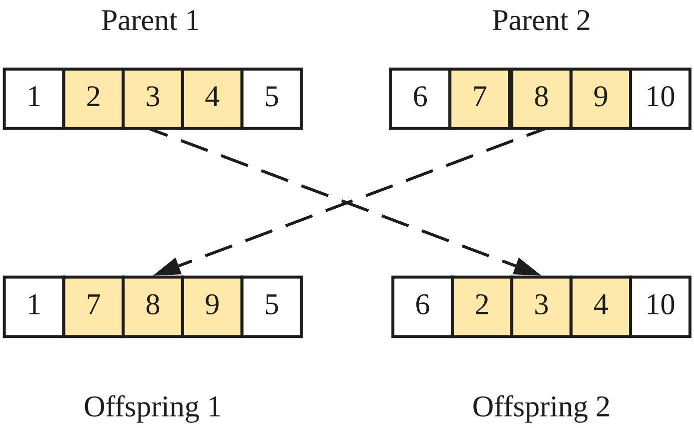


Figure 4. Two-point crossover process.

Mutation: The uniform mutation operates a replacement coming from the pool of all vehicle locations defined by street node IDs. Figure 5 illustrates the mutation operator where *Sample* is a function to randomly pick one location from the pool and $1, 2, 3, \ldots, n$ denotes all street node IDs. However, if the replacement already exists in the current solution, the process is repeated until a valid replacement is found. As mentioned before in crossover, redundant solutions will be eliminated by the process.

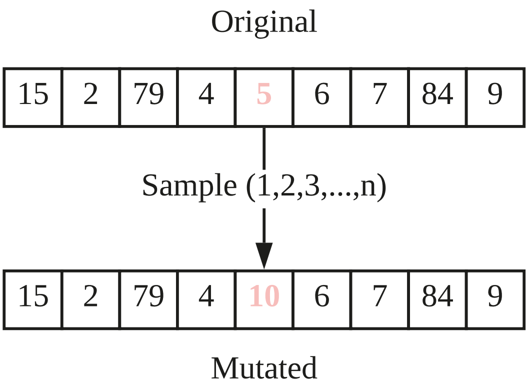


Figure 5. Uniform mutation process.

It is worth noting that evolutionary operators inherit known shortcomings such as local optimal and plateau. Evolutionary operators in this work are no exception. Several combinations of crossover and mutations have been experimented and none resulted in guaranteed superiority over all others. This work focuses on the implementation of the FPP model rather than finding suitable parameters and operators for the metaheuristic algorithm (i.e., NSGA-II) under consideration. Therefore, it may be possible that adjustment and tuning of evolutionary operators for their suitability may be necessary for its application in some instances that share too few characteristics with instances in this work.

4.4. Multi-Objective Performance Metrics

Several multi-objective quality metrics exist, which can be categorized based on the quality aspect that they assess, i.e., convergence (distance to the optima), diversity, and both convergence and diversity altogether [33]. In this work, we consider the three commonly used three indicators which measure the complementary aspects of the yielded solutions, namely, Inverted Generational Distance (IGD) [63], Spread [45] and Hypervolume (HV) [33]. Since the exact Pareto front can only be computed on small size instances, for large instances a reference front is obtained by combining the approximated Pareto fronts resulting from the heuristic and metaheuristic algorithms.

Inverted Generational Distance (IGD) [63]: This metric measures the distance between the obtained approximated solutions and the Pareto front. IGD is defined in Equation (3), where d_i is the Euclidean distance from point i in the approximated front to the closest one in the Pareto front, and n is the number of solutions in the Pareto front. $IGD = 0$ indicates that the evaluated Pareto front consists only of solutions from the optimal Pareto front.

$$IGD = \frac{\sqrt{\sum_{i=1}^{k} d_i^2}}{n} \quad . \tag{3}$$

Spread [45]: This metric measures the diversity of the obtained approximated front and is defined as:

$$\blacksquare = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (N-1)\bar{d}} \quad , \tag{4}$$

where d_i is the Euclidean distance between consecutive solutions, $\bar{d}$ is the mean of these distances, and d_f and d_l are the Euclidean distances to the *extreme* solutions of the Pareto front. A zero value indicates an ideal distribution, i.e., pointing out a perfect spread of the solutions in the evaluated set of solutions.

$$HV = \text{volume}\left(\bigcup_{i=1}^{|Q|} v_i\right) \quad . \tag{5}$$

Hypervolume [64]: This metric assesses both convergence and diversity of a Pareto front. It calculates the m-dimensional volume (in the objective space) covered by the solutions in the evaluated Pareto front Q and a dominated reference point W. For each solution $i \in Q$, a hypercube v_i is constructed with the reference point W and the solution i as the diagonal corners of the hypercube. The hypervolume is calculated as the union of all hypercubes, as shown in Equation (5). The higher the hypervolume the better the algorithm performed.

5. Execution of the Proposed Model

This section is composed of three parts. The first two parts present the process of building graphs from real street maps and the demographic data integration into the graphs. The third part describes the parameters and environment the execution.

5.1. Building Graph Instances

Most of the public street maps are usually not available in a graph format. To build graphs, street map data and footprints of the buildings are obtained from "OpenStreetMap" [65] using "OSMnx" [66] and "NetworkX" [67] tools. Then, a simplified graph is created, it combines a street map and buildings together with respect to the real position of the street nodes and buildings in V. Each edge between a building and its nearest street node is built using Open Source Routing Machine (OSRM) [54]. Edges are assigned a weight that is proportional to the walking distance extracted from OSRM. An example of a resulting instance is shown in Figure 6. Gray edges represent streets. The nodes on the streets represent street nodes, while the nodes outside the streets represent buildings in the area. Input data and parameters for all instances are summarized in Table 4.

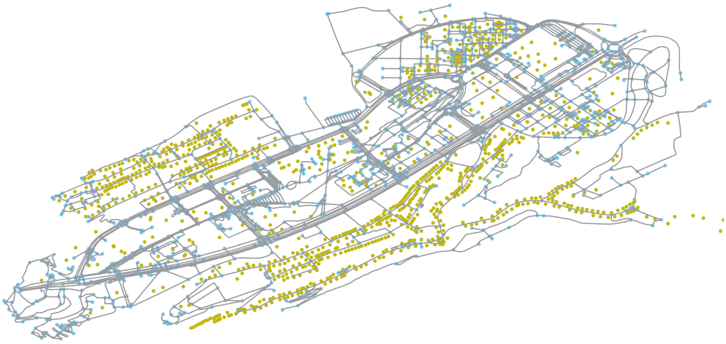


Figure 6. LU2 instance. Blue nodes represent possible locations for carsharing stations. Yellow nodes represent residential buildings.

5.2. Problem Instances

The performance of PolySCIP, heuristic algorithms (simple and iterative) and NSGA-II algorithms is compared on three real instances; LU1, LU2, and MU1 as shown in Table 4. LU1 is a small portion of Luxembourg city containing 63 street nodes and 47 buildings. LU2 consists of 2 districts of Luxembourg city that contain a total of 2026 street nodes and 1063 buildings. MU1 is an inner part of the city of Munich, which contains 16,075 street nodes and 21,816 buildings.

For the LU1 instance, the radius of the carsharing station (r) is set to zero and the maximum walking distance for users (w) is 150 m for all evaluated algorithms. The number of stations is set to four. The LU1 setup is ideal for observing and understanding the operational details of the evaluated algorithms.

The size of the LU2 graph reflects two city districts and is the smallest portion for real-world planning (deemed by the business expert). The maximum acceptable walking distance depends on the selected mode of transportation. Daniels and Mulley [53] show that people are willing to walk significantly longer to take a train than a bus as long as they deem it worthy. For carsharing, a realistic walking-to-the-car distance (w) is around 500 m [68]. The population is selected to match the real numbers reported by the city of Luxembourg [69] and is distributed uniformly in residential buildings. The area covered by a carsharing station has a radius (r) of 100 m, while the number of stations is set to 10.

The MU1 instance aims to evaluate the performance of algorithms on a city-wide scale. The population in each district is taken from municipalities and is distributed uniformly in residential buildings . We estimated the number of carsharing users at 2% of the total population. The station radius (r) and walking distance (w) are the same as in the LU2 instance, but the number of stations is increased to 100 stations. In all three instances, carsharing users are distributed uniformly among available buildings on the map. All of these settings have been defined by domain experts based on the study and real deployment plan.

Table 4. Problem instances.

	LU1	LU2	MU1
City	Luxembourg	Luxembourg	Munich
Population	561	11,439	17,486
Number of carsharing stations (f)	4	10	10,072
Number of street nodes	63	2026	16,075
Number of residential buildings	47	1063	21,816
Maximum walking distance (w)	150 m	500 m	500 m
Carsharing station area radius (r)	0 m	100 m	100 m

5.3. Algorithms Implementation and Parameters

PolySCIP (version 4.0) and heuristic algorithms are deterministic and do not require any parameters apart from those shown in Table 4. As NSGA-II is evolutionary based, it requires initial population and parameters for GA operators. Table 4 reveals the values adopted for these three instances. NSGA-II is executed for 30 times due to its stochastic nature. We develop heuristic algorithms and NSGA-II using Python 3.7 and DEAP (a library for metaheuristic algorithms) [70]. It is a common practice to include seed solutions in the initial population of metaheuristic algorithms. In this work, these are injected into the initial population as a seed solution from coverage-focused and distance-focused iterative heuristic algorithms. The configurations for NSGA-II are mentioned in Table 5.

Table 5. NSGA-II configuration parameters.

	LU1 Instance	LU2 Instance	MU1 Instance
Number of generations	400	400	400
Population size	20	50	100
Selection process	Tournament	Tournament	Tournament
Crossover method	2-point crossover	2-point crossover	2-point crossover
Crossover rate	0.8	0.9	0.9
Mutation rate	0.01	$\frac{1}{\#stations}$	$\frac{1}{\#stations}$

6. Results

The results of each instance (LU1, LU2 and MU1) are represented as a scatter plot where the x-axis represents the maximum walking distance (lower is better) and the y-axis represents the number of covered users (higher is better). Execution in this work is performed on a single core of an Intel Xeon L5640 (2.26 GHz) with courtesy of University of Luxembourg HPC.

6.1. Result of LU1 Instance

Figure 7 presents the obtained Pareto fronts from eight different algorithms and Table 6 provides the numerical results obtained for all evaluated algorithms. To simplify the table, only two extreme points on both Pareto fronts are shown. The highest achieved coverage is 391 users, which is yielded by the iterative heuristic algorithm, the exact method, and NSGA-II. On the other hand, the lowest distance of 93.5 m is achieved using PolySCIP.

Table 6. Numerical results in LU1 instance. Only extreme solutions from the two Pareto fronts are mentioned.

	Covered Users	**Maximum Walking Distance (Meters)**
PolySCIP (Best coverage)	391	149.528
PolySCIP (Best distance)	108	93.546
NSGA-II (Best coverage)	391	149.528
NSGA-II (Best distance)	203	106.4
Coverage Heuristic	348	148.491
Distance Heuristic	187	112.398
Bi-objective Heuristic	333	144.515
Coverage Iterative Heuristic	391	149.528
Distance Iterative Heuristic	87	106.4
Bi-objective Iterative Heuristic	358	144.401

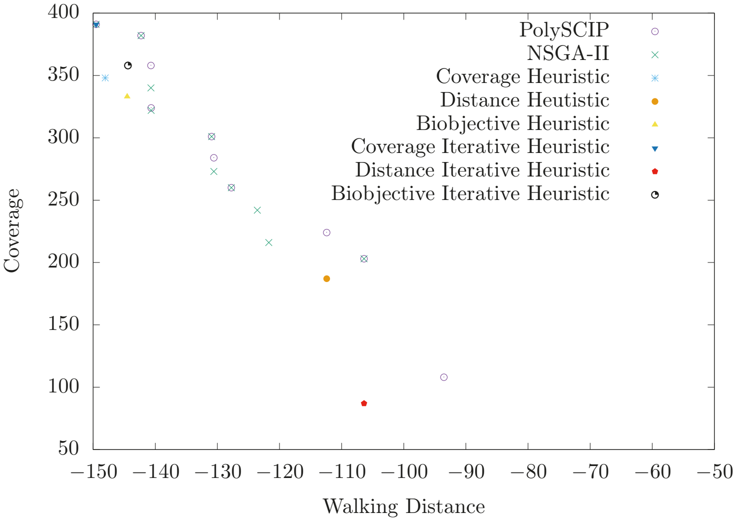


Figure 7. Results in the validation phase. The higher the value on the x axis (moving toward the right), the better the distance objective is. The higher the value on y axis, the better the user coverage objective is.

Overall, results from NSGA-II are debatably superior when both objectives are considered. The results are close to the optimum (106.4 from NSGA-II and distance-focused iterative heuristic and 93.546 m from PolySCIP). In fact, the distance result from NSGA-II is even better than its iterative counterpart as it covers more users. Figure 8 shows the stations and their respective coverage in LU1 instance.

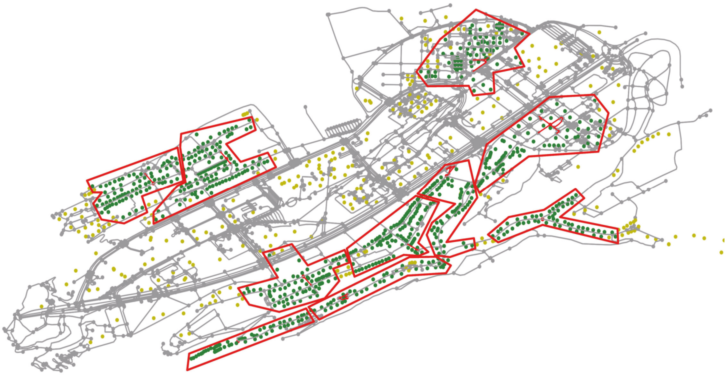


Figure 8. A map showing a solution from NSGA-II that yields the highest user coverage and the highest maximum walking distance. Covered buildings are depicted as nodes inside polygons.

For the IGD indicator (see Table 7), NSGA-II yields 3.02. This value is in accordance to Figure 7 between a true Pareto and an approximated fronts. It can be seen that NSGA-II achieved some of the solutions on the true Pareto front—especially the highest user coverage solution.

Table 7. Comparing Pareto fronts for PolySCIP and NSGA-II.

	IGD	Spread	HV
Exact method	True Pareto front	0.488	0.449
NSGA-II	3.02	0.525	0.351

As for the spread indicator, the true Pareto front yields 0.488, while the NSGA-II Pareto front yields 0.525. This means the diversity in the exact method Pareto front is better than NSGA-II's. This was due to the fact that the coverage objective overwhelmed the distance objective leading to a cluster of solutions in the upper right region in Figure 7. The hypervolume of the true Pareto front is 0.449 and the NSGA-II Pareto front yields 0.351. The difference occurs because some solutions of NSGA-II are dominated by PolySCIP's.

It is essential to note that PolySCIP was applicable for the LU1 instance due to its small size. However due to the FPP complexity, for larger instances like LU2 and MU1, PolySCIP became an enviable approach. This is elaborated in Sections 6.2 and 6.3.

6.2. Result of LU2 Instance

There is only one Pareto front from NSGA-II in Figure 9 since PolySCIP cannot deliver the solutions even after 18 days and has a memory usage of 84 GB. The plot shows that NSGA-II yields a higher coverage than the iterative heuristic coverage algorithm when it takes the iterative methods' solutions as seed solutions in the initial populations. The highest achieved user coverage is 8421 users with a maximum walking distance of 399.8 m. On the other hand, the lowest maximum walking distance achieved is 135.7 m with only 47 covered users. In Table 8, the best results from each category (i.e., simple heuristic, iterative heuristic, and NSGA-II), are compared. It reveals that NSGA-II achieves the highest user coverage and lowest walking distance among all algorithms. It can be observed in Figure 9 that even though some residential buildings are located close to carsharing stations, they are not covered. This is because the entrances of those buildings (determined during the snapping process) are mapped on the opposite streets, which are not covered by the stations. However, the number of such buildings is marginal and can be neglected.

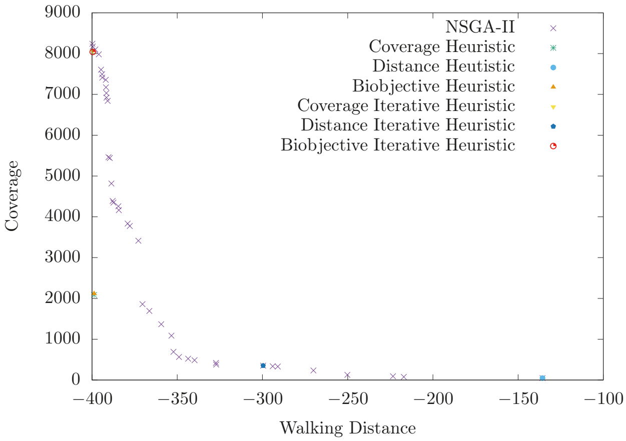


Figure 9. NSGA-II's Pareto front and heuristic algorithms' solutions for LU2. The higher the value on the x axis (moving towards the right), the better the distance objective is. The higher the value on the y axis, the better the user coverage objective is.

Table 8. Comparing best results from each algorithm categories in LU2 instance.

Algorithm	Coverage Oriented		Distance Oriented	
	Covered Users	Walking Distance	Covered Users	Walking Distance
Simple Heuristic	2100	399.8	47	135.7
Iterative Heuristic	47	135.7	231	300
NSGA-II	8421	399.8	47	135.7

6.3. Result of MU1 Instance

Due to the larger input size of the MU1 instance and FPP being NP-hard, PolySCIP cannot be employed. It takes 17 h to come up with one solution for iterative heuristic algorithms, while it takes 26 min for NSGA-II to come up with an estimated front (read Table 9). Their respective results are presented in Figure 10. The obtained results are consistent with LU1 and LU2 instances. Table 9 presents the execution time of all algorithms. Although simple heuristic algorithms take only 7 min to find a solution, the results are not comparable to the others, which are more complex. From the results, NSGA-II also achieves higher user coverage and shorter walking distance than the heuristics.

Table 9. Execution time for NSGA-II and heuristic algorithms on MU1. The measured time depicts the execution time each algorithm takes to locate 100 stations.

Algorithm	Execution Time
NSGA-II	26 min
Coverage Heuristic	7 min
Distance Heuristic	7 min
Bi-objective Heuristic	7 min
Coverage Iterative Heuristic	17 h
Distance Iterative Heuristic	17 h
Bi-objective Iterative Heuristic	17 h

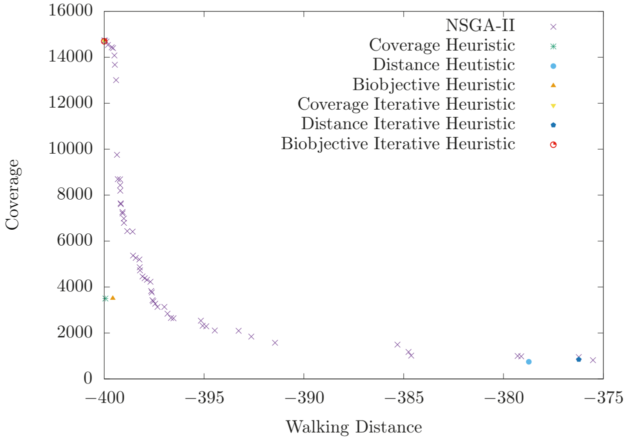


Figure 10. NSGA-II Pareto front and solutions of heuristic algorithms for the MU1 instance.

Figure 11 illustrates the NSGA-II fleet placement solution, which maximizes the number of covered users. Red pins mark locations of the carsharing stations, while green polygons show designated parking areas in inner Munich. All heuristic algorithms and NSGA-II are also compared using 72 stations in MU1 instance to compare with the manual allocation. Figure 12 shows that the manual allocation is of a lesser quality than some of the (meta-)heuristic algorithms, the iterative coverage and bi-objective version in particular, and NSGA-II. The comparison between the best results from each category of algorithm and manual allocation is also shown in Table 10. The difference in results in terms of user coverage is up to 50% (manual allocation being on the lower end), while the walking distance is similar. The results also further stress the benefit of Pareto front in decision making since it offers more options to choose from compared to the heuristic algorithms.

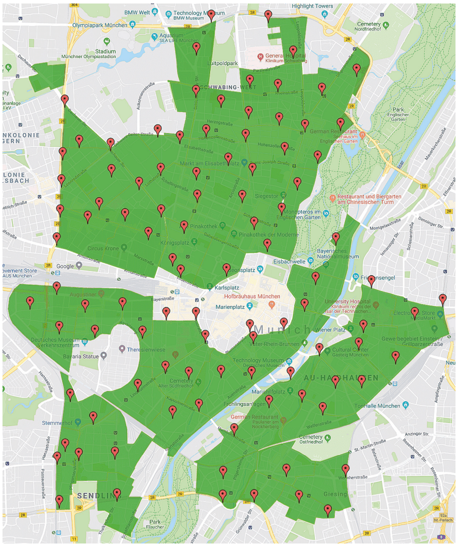


Figure 11. NSGA-II fleet placement solution which maximises user coverage in the city of Munich. Red pins are locations of the carsharing station. Green zones indicate the inner area of Munich.

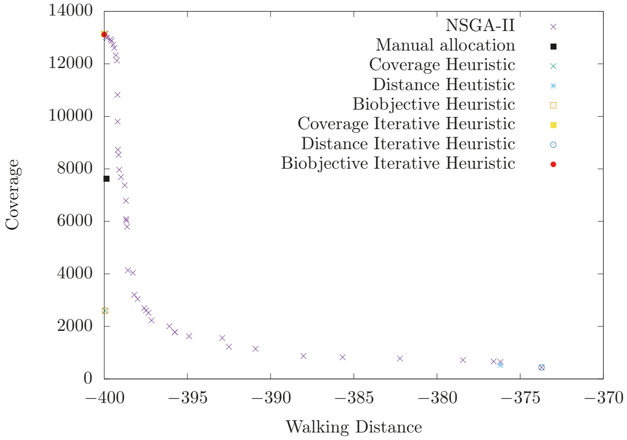


Figure 12. NSGA-II approximated Pareto front and solutions of heuristic algorithms in MU1 instance compared to the manual allocation (72 stations).

From the MU1 instance results, iterative heuristic approaches may still be possible but usually at the expense of extremely high computation time. Moreover, discovering suitable heuristics is problematic in its own. This is where the use of metaheuristic algorithms (e.g., NSGA-II) is proven to be more effective in term of solution qualities and computation time.

Table 10. Comparing the best results from each algorithm categories and manual allocation in MU1 instance.

Instance	Algorithm	Coverage Oriented		Distance Oriented	
		Covered Users	Walking Distance	Covered Users	Access Distance
Simulation	Simple Heuristic	3421	399.4	1091	378.6
	Iterative Heuristic	14,892	399.8	1214	376.1
	NSGA-II	14,892	399.8	1256	375.2
Real-world	Simple Heuristic	2291	399.8	1124	375.9
	Iterative Heuristic	13,224	399.8	986	374.2
	NSGA-II	13,224	399.8	986	374.2
	Manual Allocation	7864	399.8	7864	399.8

6.4. Discussion

Efficiency and performance: When optimality is of real concern, PolySCIP is best applied. However, its high execution time and memory requirements make it inapplicable in practice where the instance is anything greater than 1390 street nodes and 1063 buildings. PolySCIP was unable to find a solution at an acceptable time (unfinished even after 18 days), and these numbers do not represent anything close to the size of a typical large city.

Greedy algorithms, on the other hand, have the lowest execution time of all, but their results are unacceptable for practical applications due to their low user coverage. The underperformance of basic heuristic algorithms is alleviated in the iterative version, however, it comes with an additional computation cost. Despite the low execution time for a small instance, it becomes an issue in a larger instance. In the MU1 instance, the simulations took 17 h to locate 100 carsharing stations (on Intel Xeon L5640 at 2.26 GHz and over 128 GB or memory).Increasing the number of stations or increasing the size of the analyzed area will increase the execution time (in a factorial term, $n!$, where n is a number of locations) and can make it impractical.

NSGA-II's main advantages are the approximated front, the ability to cope with the size of problem instance, and the ability to improve existing solutions even further if possible. NSGA-II is 30 times faster than iterative algorithms and is still able to produce alternative solutions without needing to rerun the algorithm and change weights (in a bi-objective iterative algorithm). These properties make NSGA-II an attractive choice in finding applicable fleet placement solutions, additionally it yields a better quality solution in term of coverage than the manual allocation, which usually takes a much longer execution time.

Coverage vs. walking distance: After observing the coverage quality of distance-oriented algorithms, we found that the after the minimization of walking distance, the distance is only marginally reduced. Further analysis was carried out on the MU1 instance, with an equal weight of 0.5 to both objectives for bi-objective-focused iterative algorithm. The solution should be at (−397, 3000) in the approximated Pareto front in Figure 10. However, it is located at (−399, 14,700) instead. There are also solutions where the walking distance and user coverage are low, this is because they place stations away from crowded areas and situate the stations near a few buildings, hence, claiming the low maximum walking distance and yield lower user coverage (see Figure 13). The decrease in walking distance (in those solutions) only translates to a two to three minutes difference on foot.

The marginal difference in walking distance can be explained by the nature of the city. Users are clustered in a densely populated area. If a carsharing station is placed in such an environment, there would be users at the edge of the coverage, which makes the maximum walking distance for users to be as high as the maximum coverage distance. A station can be relocated to lower the walking distance, but the user coverage is also likely to be lower in the process. On the other hand, the walking distance can be drastically low if carsharing stations are located in an uninhabited area, but this would be detrimental to the user coverage objective and contradicts the main purpose of a carsharing service. Hence, in this work, we have shown the effect of walking distance objective in carsharing fleet placement.

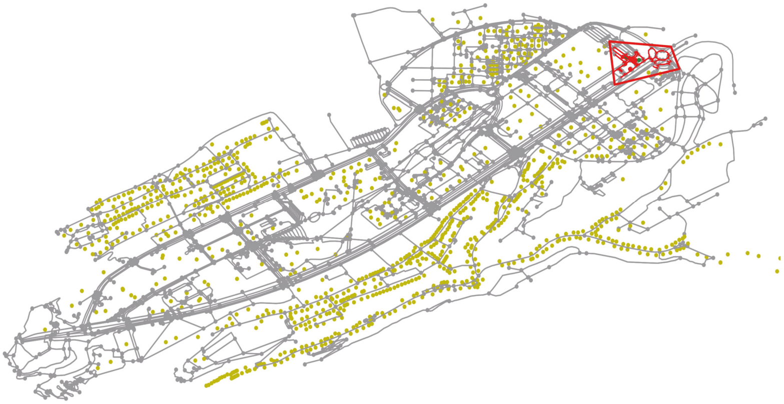


Figure 13. A solution that yields a low global walking distance, but also yields low user coverage.

7. Conclusions

This work proves that realistic Fleet Management Problem is an NP-hard problem, Apart from suggesting that exact and optimal solutions may not be realizable, it paves way to the application of heuristic and metaheuristic algorithms. This work proposed a novel methodology in optimizing fleet placement in station-based round-trip carsharing and suggests how such problems can be modeled. It is among the first to model the problem of fleet placement in carsharing and to apply a state-of-the-art optimization algorithm in attempting to determine satisfactory solutions. A set of heuristic, metaheuristic (NSGA-II), and exact (multi-criteria solver) algorithms have been applied and their performance evaluated on three instances with two objectives, i.e.,maximizing the number of carsharing users and minimizing the maximum global walking-to-the-car distance, under consideration.

This work is also the first to use real and exact instances (instead of just an abstract) for the study. Three different instances have been used, and each has its own characteristics with different sizes and objectives to reflect real world demand. The proposed method demonstrates that NSGA-II is superior to the manual allocation by a significant margin in user coverage and in terms of approximated Pareto front, and presents a number of solutions for decision makers to choose from. Solutions from our proposed method are also more efficient in terms of both user coverage and walking distance. While a metaheuristic approach has received much attention lately, this works affirms its application in transportation and Fleet Management Problem, in particular. The model proposed ought to be a good starting point in solving similar problems for further research among transportation and logistic communities.

Future work could apply the proposed approach to other cities such as London, Athens, and Paris. Additional objectives such as car fleet utilization, car fleet size, and the number of stations may also be included, since these are also real concerns after the initial launch of the carsharing business. A tailor-made metaheuristic algorithm may also be invented with the Fleet Management problem in mind too.

Author Contributions: Conceptualization, B.C., D.K. and P.B.; methodology, B.C., Grégoire Danoy, M.B. and F.G.; software, B.C.; validation, B.C., J.M. and K.L.; formal analysis, B.C., F.G. and J.M.; investigation, B.C. and D.K.; resources, D.K. and P.B.; data curation, B.C.; writing—original draft preparation, B.C., G.D. and D.K.; writing—review and editing, B.C. and K.L.; visualization, B.C.; supervision, P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. CarsharingNews. Carsharing Anbieter. 2019. Available online: https://www.carsharing-news.de/carsharing-anbieter/ (accessed on 3 April 2019).
2. Deloitte. Car Sharing in Europe: Business Models, National Variations and Upcoming Disruptions. 2017. Available online: https://www2.deloitte.com/content/dam/Deloitte/de/Documents/consumer-industrial-products/CIP-Automotive-Car-Sharing-in-Europe.pdf (accessed on 29 August 2017).
3. Luè, A.; Colorni, A.; Nocerino, R.; Paruscio, V. Green move: An innovative electric vehicle-sharing system. *Procedia-Soc. Behav. Sci.* **2012**, *48*, 2978–2987. [CrossRef]
4. Reichert, A.; Holz-Rau, C. Mode use in long-distance travel. *J. Transp. Land Use* **2015**, *8*, 87–105. [CrossRef]
5. Church, R.; ReVelle, C. The maximal covering location problem. In *Papers of the Regional Science Association*; Springer: Berlin/Heidelberg, Germany, 1974; Volume 32, pp. 101–118.
6. Megiddo, N.; Supowit, K.J. On the complexity of some common geometric location problems. *SIAM J. Comput.* **1984**, *13*, 182–196. [CrossRef]
7. Seargeant, D.B. The Maximal Covering Location Problem: An Application in Reproductive Health Services. Ph.D. Thesis, University of California, Los Angeles, CA, USA, 2012.
8. Schmid, V.; Doerner, K.F. Ambulance location and relocation problems with time-dependent travel times. *Eur. J. Oper. Res.* **2010**, *207*, 1293–1303. [CrossRef] [PubMed]
9. Ghaffarinasab, N.; Motallebzadeh, A. Hub interdiction problem variants: Models and metaheuristic solution algorithms. *Eur. J. Oper. Res.* **2018**, *267*, 496–512. [CrossRef]
10. Xiao, N.; Bennett, D.A.; Armstrong, M.P. Using evolutionary algorithms to generate alternatives for multiobjective site-search problems. *Environ. Plan. A* **2002**, *34*, 639–656. [CrossRef]
11. Kim, K.; Murray, A.T. Enhancing spatial representation in primary and secondary coverage location modeling. *J. Reg. Sci.* **2008**, *48*, 745–768. [CrossRef]
12. Malekpoor, H.; Chalvatzis, K.; Mishra, N.; Ramudhin, A. A hybrid approach of vikor and bi-objective integer linear programming for electrification planning in a disaster relief camp. *Ann. Oper. Res.* **2018**, *283*, 443–469. [CrossRef]
13. Megiddo, N.; Tamir, A. On the complexity of locating linear facilities in the plane. *Oper. Res. Lett.* **1982**, *1*, 194–197. [CrossRef]
14. Laporte, G.; Nickel, S.; da Gama, F.S. *Location Science*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 528.

15. Sakai, T.; Kawamura, K.; Hyodo, T. The relationship between commodity types, spatial characteristics, and distance optimality of logistics facilities. *J. Transp. Land Use* **2018**, *11*, 575–591. [CrossRef]
16. Razi, N.; Karatas, M. A multi-objective model for locating search and rescue boats. *Eur. J. Oper. Res.* **2016**, *254*, 279–293. [CrossRef]
17. Raisanen, L.; Whitaker, R.M. Comparison and evaluation of multiple objective genetic algorithms for the antenna placement problem. *Mob. Netw. Appl.* **2005**, *10*, 79–88. [CrossRef]
18. Vasquez, M.; Hao, J.K. A heuristic approach for antenna positioning in cellular networks. *J. Heuristics* **2001**, *7*, 443–472. [CrossRef]
19. Awasthi, A.; Breuil, D.; Chauhan, S.S.; Parent, M.; Reveillere, T. A multicriteria decision making approach for carsharing stations selection. *J. Decis. Syst.* **2007**, *16*, 57–78. [CrossRef]
20. Kumar, P.; Bierlaire, M. Optimizing locations for a vehicle sharing system. In Proceedings of the Swiss Transport Research Conference, Number EPFL-CONF-195890, Ascona, Switzerland, 2–4 May 2012.
21. Boyacı, B.; Zografos, K.G.; Geroliminis, N. An optimization framework for the development of efficient one-way car-sharing systems. *Eur. J. Oper. Res.* **2015**, *240*, 718–733. [CrossRef]
22. Lage, M.; Machado, C.; Berssaneti, F.; Quintanilha, J. A method to define the spatial stations location in a carsharing system in São Paulo–Brazil. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *XLII-4/W11*, 27–32. [CrossRef]
23. Schwer, J.; Timpf, S. Local-Level Site-Selection Model for Integrated Carsharing Services. *GI_Forum* **2016**, *4*, 243–249. [CrossRef]
24. Calık, H.; Fortz, B. Location of stations in a one-way electric car sharing system. In Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 3–6 July 2017; pp. 134–139.
25. Jiao, Z.; Ran, L.; Chen, J.; Meng, H.; Li, C. Data-driven approach to operation and location considering range anxiety of one-way electric vehicles sharing system. *Energy Procedia* **2017**, *105*, 2287–2294. [CrossRef]
26. Brandstätter, G.; Kahr, M.; Leitner, M. Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand. *Transp. Res. Part B Methodol.* **2017**, *104*, 17–35. [CrossRef]
27. Yıldız, B.; Olcaytu, E.; Şen, A. The urban recharging infrastructure design problem with stochastic demands and capacitated charging stations. *Transp. Res. Part B Methodol.* **2019**, *119*, 22–44. [CrossRef]
28. Datner, S.; Raviv, T.; Tzur, M.; Chemla, D. Setting inventory levels in a bike sharing network. *Transp. Sci.* **2017**, *53*, 62–76. [CrossRef]
29. García-Palomares, J.C.; Gutiérrez, J.; Latorre, M. Optimizing the location of stations in bike-sharing programs: A GIS approach. *Appl. Geogr.* **2012**, *35*, 235–246. [CrossRef]
30. Liu, J.; Li, Q.; Qu, M.; Chen, W.; Yang, J.; Xiong, H.; Zhong, H.; Fu, Y. Station site optimization in bike sharing systems. In Proceedings of the 2015 IEEE International Conference on Data Mining, Atlantic City, NJ, USA, 14–17 November 2015; pp. 883–888.
31. Raviv, T.; Tzur, M.; Forma, I.A. Static repositioning in a bike-sharing system: Models and solution approaches. *EURO J. Transp. Logist.* **2013**, *2*, 187–229. [CrossRef]
32. Romero, J.P.; Ibeas, A.; Moura, J.L.; Benavente, J.; Alonso, B. A simulation-optimization approach to design efficient systems of bike-sharing. *Procedia-Soc. Behav. Sci.* **2012**, *54*, 646–655. [CrossRef]
33. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74.
34. Efroymson, M.; Ray, T. A branch-bound algorithm for plant location. *Oper. Res.* **1966**, *14*, 361–368. [CrossRef]
35. Manopiniwes, W.; Irohara, T. Stochastic optimisation model for integrated decisions on relief supply chains: Preparedness for disaster response. *Int. J. Prod. Res.* **2017**, *55*, 979–996. [CrossRef]
36. Zarandi, M.F.; Davari, S.; Sisakht, S.H. The large scale maximal covering location problem. *Sci. Iran.* **2011**, *18*, 1564–1570. [CrossRef]
37. CPLEX, I.I. V12. 1: User's Manual for CPLEX. *Int. Bus. Mach. Corp.* **2009**, *46*, 157.
38. Galvão, R.D.; ReVelle, C. A Lagrangean heuristic for the maximal covering location problem. *Eur. J. Oper. Res.* **1996**, *88*, 114–123. [CrossRef]
39. Gendron, B.; Khuong, P.V.; Semet, F. A Lagrangian-based branch-and-bound algorithm for the two-level uncapacitated facility location problem with single-assignment constraints. *Transp. Sci.* **2016**, *50*, 1286–1299. [CrossRef]
40. Gendreau, M.; Laporte, G.; Semet, F. Solving an ambulance location model by tabu search. *Locat. Sci.* **1997**, *5*, 75–88. [CrossRef]
41. Murray, A.T.; Church, R.L. Applying simulated annealing to location-planning models. *J. Heuristics* **1996**, *2*, 31–53. [CrossRef]
42. Ho, S.C. An iterated tabu search heuristic for the single source capacitated facility location problem. *Appl. Soft Comput.* **2015**, *27*, 169–178. [CrossRef]
43. Ye, F.; Zhao, Q.; Xi, M.; Dessouky, M. Chinese National Emergency Warehouse Location Research based on VNS Algorithm. *Electron. Notes Discret. Math.* **2015**, *47*, 61–68. [CrossRef]
44. Karasakal, E.; Silav, A. A multi-objective genetic algorithm for a bi-objective facility location problem with partial coverage. *Top* **2016**, *24*, 206–232. [CrossRef]
45. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
46. Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **2000**, *8*, 173–195. [CrossRef]
47. RanjbarTezenji, F.; Mohammadi, M.; Pasandideh, S.H.R. Bi-objective location-allocation-inventory-network design in a two-echelon supply chain using de novo programming, NSGA-II and NRGA. *Int. J. Logist. Syst. Manag.* **2017**, *28*, 308–337. [CrossRef]

48. de Almeida Correia, G.H.; Antunes, A.P. Optimization approach to depot location and trip selection in one-way carsharing systems. *Transp. Res. Part E Logist. Transp. Rev.* **2012**, *48*, 233–247. [CrossRef]
49. Asamer, J.; Reinthaler, M.; Ruthmair, M.; Straub, M.; Puchinger, J. Optimizing charging station locations for urban taxi providers. *Transp. Res. Part A Policy Pract.* **2016**, *85*, 233–246. [CrossRef]
50. Mourad, A.; Puchinger, J.; Chu, C. A survey of models and algorithms for optimizing shared mobility. *Transp. Res. Part B Methodol.* **2019**, *123*, 323–346. [CrossRef]
51. Barth, M.; Todd, M. Simulation model performance analysis of a multiple station shared vehicle system. *Transp. Res. Part C Emerg. Technol.* **1999**, *7*, 237–259. [CrossRef]
52. Ströhle, P.; Flath, C.M.; Gärttner, J. Leveraging customer flexibility for car-sharing fleet optimization. *Transp. Sci.* **2018**, *53*, 42–61. [CrossRef]
53. Daniels, R.; Mulley, C. Explaining walking distance to public transport: The dominance of public transport supply. *J. Transp. Land Use* **2013**, *6*, 5–20. [CrossRef]
54. Luxen, D.; Vetter, C. Real-time routing with OpenStreetMap data. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '11, Chicago, IL, USA, 1–4 November 2011; pp. 513–516.
55. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; W. H. Freeman & Co.: New York, NY, USA, 1979.
56. Karp, R.M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; Springer: Berlin/Heidelberg, Germany, 1972; pp. 85–103.
57. Raz, R.; Safra, S. A Sub-constant Error-probability Low-degree Test, and a Sub-constant Error-probability PCP Characterization of NP. In Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, STOC '97, El Paso, TX, USA, 4–6 May 1997; ACM: New York, NY, USA, 1997; pp. 475–484. [CrossRef]
58. Borndörfer, R.; Schenker, S.; Skutella, M.; Strunk, T. PolySCIP. In *International Congress on Mathematical Software*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 259–264.
59. Fourer, R.; Gay, D.M.; Kernighan, B. *Ampl*; Boyd & Fraser: Danvers, MA, USA, 1993; Volume 117.
60. Laumanns, M.; Thiele, L.; Zitzler, E. Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. *Nat. Comput.* **2004**, *3*, 37–51. [CrossRef]
61. Corne, D.W.; Knowles, J.D.; Oates, M.J. The Pareto envelope-based selection algorithm for multiobjective optimization. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Paris, France, 18–20 September 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 839–848.
62. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [CrossRef]
63. Van Veldhuizen, D.A. Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Ph.D. Thesis, Wright Patterson AFB, Montgomery County, OH, USA, 1999.
64. Zitzler, E.; Thiele, L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [CrossRef]
65. OpenStreetMap Contributors. Luxembourg City and Munich City Dump. 2017. Available online: https://www.openstreetmap.org (accessed on 28 June 2018).
66. Boeing, G. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Comput. Environ. Urban Syst.* **2017**, *65*, 126–139. [CrossRef]
67. Hagberg, A.A.; Schult, D.A.; Swart, P.J. Exploring network structure, dynamics, and function using NetworkX. In Proceedings of the 7th Python in Science Conference (SciPy2008), Pasadena, CA, USA, 19–24 August 2008; pp. 11–15.
68. Frost and Sullivan. Future of Carsharing Market to 2025. 2016. Available online: https://store.frost.com/future-of-carsharing-market-to-2025.html (accessed on 29 August 2017).
69. La Ville de Luxembourg. Etat de la Population 2019. 2016. Available online: https://statistiques.public.lu/stat/ReportFolders/ReportFolder.aspx?IF_Language=fra&MainTheme=2&FldrName=1 (accessed on 15 August 2017).
70. Fortin, F.A.; De Rainville, F.M.; Gardner, M.A.; Parizeau, M.; Gagné, C. DEAP: Evolutionary Algorithms Made Easy. *J. Mach. Learn. Res.* **2012**, *13*, 2171–2175.

applied sciences

Article

Stock Portfolio Management in the Presence of Downtrends Using Computational Intelligence

Raymundo Díaz [1], Efrain Solares [2,*], Victor de-León-Gómez [2] and Francisco G. Salas [2]

[1] Business School, Tecnologico de Monterrey, Monterrey 64849, Mexico; raymundo.diaz@tec.mx
[2] Faculty of Accounting and Administration, Universidad Autónoma de Coahuila, Torreón 27000, Mexico; v.gomez@uadec.edu.mx (V.d.-L.-G.) francisco.salas@uadec.edu.mx (F.G.S.)
* Correspondence: efrain.solares@uadec.edu.mx

Abstract: Stock portfolio management consists of defining how some investment resources should be allocated to a set of stocks. It is an important component in the functioning of modern societies throughout the world. However, it faces important theoretical and practical challenges. The contribution of this work is two-fold: first, to describe an approach that comprehensively addresses the main activities carried out by practitioners during portfolio management (price forecasting, stock selection and portfolio optimization) and, second, to consider uptrends and downtrends in prices. Both aspects are relevant for practitioners but, to the best of our knowledge, the literature does not have an approach addressing them together. We propose to do it by exploiting various computational intelligence techniques. The assessment of the proposal shows that further improvements to the procedure are obtained when considering downtrends and that the procedure allows obtaining portfolios with better returns than those produced by the considered benchmarks. These results indicate that practitioners should consider the proposed procedure as a complement to their current methodologies in managing stock portfolios.

Keywords: short selling; stock portfolios; artificial neural networks; evolutionary algorithms; computational intelligence

Citation: Díaz, R.; Solares, E.; de-León-Gómez, V.; Salas, F.G. Stock Portfolio Management in the Presence of Downtrends Using Computational Intelligence. *Appl. Sci.* **2022**, *12*, 4067. https://doi.org/10.3390/app12084067

Academic Editors: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Received: 24 February 2022
Accepted: 14 April 2022
Published: 18 April 2022

1. Introduction

Both individual and organizational investors commonly seek to take profits from stock markets. Among the different ways to exploit these markets, the literature has focused on the idea of buying cheap and selling expensive. The authors of [1] point out that there is an assumption in classical portfolio theory to manage the selected assets with the simplest trading strategy, which is a buy-and-hold approach. However, it is also common for practitioners to also seek profits when prices go down. There are several mechanisms that allow an investor to take profits in this situation (e.g., [2–4]).

Investing in stocks when their prices are expected to rise is known as opening a long position. In this scenario, the investor adopts the idea that stocks should be bought when they are the cheapest and sold when they are as expensive as possible; the difference between selling and buying prices constitutes the investor's basic earning. On the other hand, opening a short position means that the investor expects the stock prices to go down. According to [5], short selling allows the investor to profit from their belief that the price of a security will decline. Moreover, short selling is used by top-down and quantitative managers as a part of a neutral strategy (cf. [5]). In this case, the investor can, for example, borrow shares of the stock, sell them in this very moment and commit to return them at a moment in the future; so, to return them, the investor will have to buy them at whatever the price of the stock is at that moment in the future. Therefore, the earning of the investment here is also calculated as the difference between the selling and buying prices—just that the sell is produced first.

The highly complex decision-making process of allocating resources considering both uptrends and downtrends of prices requires sophisticated models and tools to achieve competitive results. Thus, this work proposes a comprehensive procedure based on computational intelligence that aids defining how investors should allocate their resources in the presence of both scenarios.

First, an artificial neural network (ANN) (cf. [6]) is used to estimate future prices. There are evident tendencies in the literature showing that ANNs have high accuracy, fast prediction speed and clear superiority in predictions related to financial markets (e.g., [7–9]). To perform these estimations, the ANN takes historical performances of the stocks considering the most common factors of the literature, such as stock prices and financial ratios (cf. [10]). Some additional financial indicators are used here to determine if the forecasted tendency (that the price will go up or down) is supported. These indicators are taken from the so-called fundamental analysis, a type of indicators often considered by practitioners (cf. [11]). Evolutionary algorithms (EAs) are then used to ponder these indicators altogether with the price estimation and determine which stocks should be considered by the investor for investment, either with a downtrend or an uptrend. Finally, EAs are also used to determine how much of the resources should be allocated to each of the selected stocks on the basis of statistical analysis to historical data. Here, only historical prices of the selected stocks are taken into consideration according to the approach described in [12].

The literature review presented in Section 3 shows that, although there are studies that consider both uptrends and downtrends in stock prices, as far as we know, there are no published works that comprehensively address the problem the way that is proposed here. That is, not only taking advantage of a future increase in prices by opening long positions but also taking advantage of future decrease in prices by opening short positions, while also forecasting stock prices, selecting the most plausible stocks and optimizing the stock portfolio. Our hypothesis is that a procedure that effectively implements all this provides better overall earnings for the investor. The hypothesis is based on the activities and interests of practitioners. We test this hypothesis by using extensive experiments with actual historical data.

The rest of the paper is structured as follows. Section 2 describes the fundamental theories that support this research. Section 3 presents the related literature. Section 4 describes the details of the techniques that compose the proposed procedure. In Section 5, we explained the experiments to test this work's hypothesis. Finally, Section 6 concludes this paper.

2. Background

This section provides a brief overview of the concepts and methods used in the proposed approach. These concepts and methods are (i) fundamental analysis, (ii) artificial neural networks and (iii) evolutionary multi-objective optimization. Furthermore, we provide a short description of multiobjective optimization problems in order to present a complete theoretical basis of the proposed approach.

2.1. Fundamental Analysis

One of the most used sources of information in the management of stock portfolios comes from the so-called fundamental analysis. The fundamental indicators provided by this analysis allow the practitioner to evaluate stocks from multiple perspectives. Such indicators are constructed from the financial statements that the companies (underlying the stocks) present publicly on a regular basis.

Fundamental indicators provide information that is often exploited in the literature to forecast future stock performance and to select the most competitive stocks. These indicators can be used both qualitatively and quantitatively. Regarding the latter, the financial information published by companies is synthesized in the form of ratios that shed light on the current state of the company, providing remarkable information on what can be expected from the financial health of the company and the possible future price of its stock. When this analysis is used in the literature, the fundamental indicators are usually aggregated in an overall assessment value that requires subjective preferences from the practitioner (cf. e.g., [13]); however, the aggregation procedure is not straightforward and represents an important challenge.

On the other hand, different fundamental indicators could be more convenient for companies with different types of activities ([14]). Some fundamental indicators that can be used for trans-business companies are described in Section 4.1 (cf. [10,11,13,14]).

2.2. Artificial Neural Networks

Artificial neural networks are nowadays very popular among techniques from computational intelligence that have been used for many applications, such as classification, clustering, pattern recognition and prediction in diverse scientific and technological disciplines ([15,16]). Similarly to other computational intelligence techniques, applications of ANN are very diversified due to its capability to model systems and phenomena from the fields of sciences, engineering and social sciences.

Analogously to a nervous system, an ANN is built from neurons, which are the basic elements for processing signals. Neurons are interconnected to form a network, with additional connections (synaptic relations) for input and output signals. Weights are assigned to each of these and other connection. The computing of suitable values for these weights is performed by training algorithms. An ANN needs to be trained before it can be used by using data from the system or phenomenon to model. Neurons are configured to form layers, in which neurons have parallel connections for inputs and outputs. ANN complexity varies from a network with a single layer of a single neuron to networks with several layers, each having several neurons. Networks with only forward connections are known as feedforward networks. Networks with forward and backward connections are known as feedbackward networks ([15]). The term deep learning refers to ANN with complex multilayers ([17]). Roughly speaking, deep learning has more complex connections between layers and also more neurons than previous types of networks. Some neural networks that form deep learning networks are convolutional networks, recursive networks and recurrent networks.

2.3. Multi-Objective Optimization Problem

Without loss of generality, a multi-objective optimization problem (MOP) can be defined in terms of maximization (although minimization is also common) as follows:

$$\text{maximize } F(x) = [f_1(x), f_2(x), \ldots, f_k(x)]^\top$$

$$\text{subject to } x \in \Omega$$

where Ω is the set of decision variable vectors $x = [x_1, x_2, \ldots, x_m)]^\top$ that fulfill the set of constraints of the problem, and then $F : \Omega \to \mathbb{R}^k$, where $\mathbb{R}^k$ is the so-called objective space.

It is evident that the notation used here states that all functions f_i (objectives) should be maximized; however, it is also possible that one requires some functions f_i to be minimized instead. To keep standard notation, we assume that the latter can be simply achieved by multiplying the minimizing function by -1.

In the context of stock portfolio management, the functions f_i are usually in conflict with each other. This means that improving f_j deteriorates f_k for some $j \neq k$. Therefore, there is no solution $x \in \Omega$ that maximizes all the k objectives simultaneously. Nevertheless, it is still possible to define some solutions x that poses the best characteristics in terms of their impact on the objectives; this is commonly carried out through Pareto optimality (cf. [18]).

Let u, vs. $\in \mathbb{R}^k$ denote impacts of solutions x and y, respectively. u dominates v if and only if $u_i \geq v_i$ for all $i = 1, \ldots, k$, and $u_j > v_j$ for at least one $j = 1, \ldots, k$. Then, a solution $x^* \in \Omega$ is Pareto optimal if there is no solution $y \in \Omega$ such that $F(y)$ dominates $F(x^*)$. Note that there can be more than one Pareto optimal solution. The set of all the Pareto optimal solutions is called the Pareto set (PS), and the set of all their corresponding objective vectors is the Pareto front (PF).

2.4. Evolutionary Multi-Objective Optimization

Multi-objective evolutionary algorithms (MOEAs) are high-level procedures designed to discover good enough solutions to MOPs (solutions that are close to the global optimum). They are especially useful with incomplete or imperfect information or a limited computing capacity ([19]).

MOEAs address MOPs using principles from biological evolution. They use a population of individuals, each representing a solution to the MOP. The individuals in the population reproduce among them, using so-called evolutionary operators (selection, crossover, mutation), to produce a new generation of individuals. Often, this new generation of individuals is composed of both parents and children that posses the best fitness; this fitness represents the impact on the objectives of the MOP. Since each individual encodes a solution to the MOP, MOEAs can approximate a set of trade-off alternatives simultaneously

The performance of MOEAs has been assessed in different fields (e.g., [20,21]). They have been widely accepted as convenient tools for addressing the problem of stock portfolio management ([10–12]). The main goal of MOEAs is to find a set of solutions that approximate the true Pareto front in terms of convergence and diversity. Convergence refers to determining the solutions that belong to the PF, while diversity refers to determining the solutions that best represent all the PF. Thus, the intervention of the decision maker is not traditionally used in the process. Thus, rather little interest has been paid in the literature to choosing one of the efficient solutions as the final one in contrast to the interest paid in approximating the whole Pareto front.

Usually, two types of MOEAs are highlighted in the literature: differential evolution and genetic algorithms. Differential evolution (DE) has been found to be very simple and effective ([22]), particularly when addressing non-linear single-objective optimization problems ([23,24]). On the other hand, in a genetic algorithm (GA), solutions to a problem are sought in the form of strings of characters (the best representations are usually those that reflect something about the problem that is being addressed), virtually always applying recombination operators such as crossing, selection and mutation operators. GAs compose one of the most popular meta-heuristics applied to the Portfolio Optimization Problem ([12]).

As a very effective and efficient way to address MOPs, the authors of ([25]) exploited the idea of creating subproblems underlying the original optimization problem. This way, addressing these subproblems the algorithm proposed in ([25]) indirectly addresses the original problem. In that work, the so-called Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) was presented. The goal of MOEA/D is to create subproblems such that, for each subproblem, a simpler optimization problem can be more effectively and efficiently addressed; each subproblem consists on the aggregation of all the objectives through a scalar function. MOEA/D was extended to the context of interval numbers in [12].

3. Literature Review

There are many contributions to portfolio management literature in recent years. In this section, we give an overview of some recent and relevant works on the following subjects: price forecasting, stock selection and portfolio optimization, as well as works on portfolio management by algorithms that exploits both uptrends and downtrends in stock prices.

Due to the non-linearity of stock data, a model developed using traditional approaches with single intelligent techniques may not use the resources in an effective way. Therefore, there is a need for developing a hybridization of intelligent techniques for an effective predictive model [26].

3.1. Portfolio Management: Price Forecasting, Stock Selection and Portfolio Optimization

In recent years, there have been plenty of contributions on price forecasting based on either statistical or computational intelligence methods (see [10,27]). The stock market is characterized by extreme fluctuations, non-linearity, and shifts in internal and external environmental variables. Artificial intelligence techniques can detect such non-linearity, resulting in much-improved forecast results [28].

Among the computational intelligence methods used for price forecasting are deep learning (e.g., [29–32]) and machine learning (e.g., [33–35]). In [10], a hybrid stock selection model with a stock prediction stage based on an artificial neural network (ANN) trained with the extreme learning machine (ELM) training algorithm ([6,36]) was proposed. The ELM algorithm has been tested for financial market prediction in other works (see [7–9]).

There are important works on methods for stock selection, which have several different fundamental theories, from operations research methods (e.g., [37,38]) to approaches originating in modern portfolio theory (Mean-variance model) (e.g., [38,39]) and soft computing methods (e.g., [40,41]), including hybrid approaches (e.g., [10,42,43]).

The fundamental theory for portfolio optimization is Markowitz's mean-variance model ([44]). Its formulation marked the beginning of Modern portfolio theory ([45]). However, Markowitz's original model is considered too basic since it neglects real-world issues related to investors, trading limitations, portfolio size and others ([43]). For evaluating a portfolio's performance, the model is based on measuring the expected return and the risk; the latter is represented by the variance in the portfolio's historical returns. Since the variance takes into account both negative and positive deviations, other risk measures have been proposed, such as the Conditional Value at Risk (CVaR) ([46,47]). As a result, numerous works have improved the model, creating more risk measures and proposing restrictions that bring them closer to practical aspects of stock market trading ([27]). Consequently, many optimization methods based on exact algorithms (e.g., [48–55]) and heuristic and hybrid optimization (e.g., [29,56–65]) have been proposed to solve the emerging portfolio optimization models ([27,40,45]).

According to [12], the investor or decision maker in the portfolio selection problem manages a multiple criteria problem in which, along with the objective of return maximization, he/she faces the uncertainty of risk. Different attitudes assumed by decision makers may lead them to select different alternatives. A way of modeling both risk and subjectivity of the decision maker in terms of significant confidence intervals was first proposed in [12]. The probabilistic confidence intervals of the portfolio returns characterize the portfolios during the optimization. The optimization is performed by means of a widely accepted decomposition-based evolutionary algorithm, the MOEA/D ([25,66]). This approach is inspired on the independent works of ([67,68]) on interval analysis theory.

3.2. Exploiting Uptrends and Downtrends in Strategies for Stock Investment

Regarding alternative strategies to the known buy-and-hold approach for stock investment, in ([69]), the authors propose two new trading strategies to outperform the buy-and-hold approach, which is based on the efficient market hypothesis. The proposed strategies are based on a generalized time-dependent strategy proposed in ([70]) but propose different timing for changing the buying/selling position. According to ([71]), the decision to adopt a long or short position in an asset requires a view of its immediate future price movements. A typical short seller would have to assess the potential future behavior of the asset price by means of evaluating several factors, such as past returns and market effects as well as and technical indicators, such as market ratios ([71]). There are a few works published in the literature to address the problem of trading strategies for the short position. An interesting work that considers not only the short position but both the short and long position is ([72]), in which a simultaneous long-short trading strategy (SLS) is proposed. Such a strategy is based partially on the property that a positive gain with zero initial investment is expected, which holds for all discrete and continuous price processes with independent multiplicative growth and a constant trend. Other works based on SLS are ([73–75]). However, these works show the results of the algorithm on a previously defined stock portfolio, unlike the proposed approach that comprehensively performs price forecasting, stock selection and portfolio optimization in the presence of both uptrends and downtrends.

4. Methods and Materials

The procedure followed here consists of applying several techniques from the so-called computational intelligence to address the complexity of stock investments in the presence of both increasing and decreasing prices. Future stock prices are forecasted using an ANN, as well as the tendency that such prices will show. These estimations are then combined with certain indicators from the fundamental analysis to define the stocks that will likely receive resources (the selected stocks). Finally, another evolutionary algorithm is used to optimize portfolios, i.e., to define the proportions of resources to be allocated to each stock.

4.1. An Artificial Neural Network to Estimate Future Prices

In this work, the immediate next period price of the considered stocks are estimated by means of an ANN. Following the recommendations of ([6,10,36]), we use a single-layer feedforward network (whose setting is created once per each stock) and train the ANN by means of the so-called extreme learning machine algorithm because of its superior capacities in similar problems to the one addressed here (cf. [7–9]).

The ANN works independently per stock to estimate its price in the subsequent immediate period. The return of each stock is used as the target variable, while thirteen variables are used as input to train the ANN. Let r_t denote the stock return for a given period t. r_t is calculated from the stock price for that period (p_t) and the immediate previous one (p_{t-1}), as defined by Equation (1).

$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}} \tag{1}$$

The high complexity involved in forecasting future stock prices requires one to consider a variety of transaction data as explanatory variables. Therefore, we followed the recommendations provided in ([10,76,77]) to determine sixteen transaction data as explanatory variables to the forecasting model used here. The sixteen input variables are described as follows:

Close price. Last transacted price of the stock before the market officially closes.
Open Price. First price of the stock at which it was traded at the open of the period's trading.
High. Highest price of the stock in the period's trading.
Low. Lowest price of the stock in the period's trading.
Average Price. Average price of the stock in the period's trading.
Market Capitalization. Price per share multiplied by the number of outstanding shares of a publicly held company.
Return Rate. Profit on an investment over a period, expressed as a proportion of the original investment.
Volume. Number of shares traded (or their equivalent in money) of a stock in a given period.
Total asset turnover. Net sales over the average value of total assets on the company's balance sheet between the beginning and the end of the period.
Fixed asset turnover. Net sales over the average value of fixed assets.
Volatility. Standard deviation of prices.
General Capital. Number of preferred and common shares that a company is authorized to issue.
Price to Earnings. Market value per share over earnings per share.
Price to Book. Market price per share over book value per share.
Price to Sales. Market price per share over revenue per share.

Price to Cash Flow. Market price per share over operating cash flow per share.

The training process consists of taking sixty historical values for these sixteen variables randomly out of a set of ninety historical periods and leaving the rest of values to test the ANN. After the ANN is trained, two errors are computed: training error and testing error. The lower the testing error, the better the predictive capacity the ANN has. Nevertheless, since the extreme learning machine algorithm uses a random procedure to compute the weights and bias of the network, we do not always obtain the same results. Therefore, we run the algorithm n_a times and chose the one with better results. It is important to highlight that each input variable is normalized taking into account the sixty periods of the training data (the target variable is not normalized).

As mentioned before, our approach seeks to take advantage of market downtrends. To achieve this, we use the ANN's forecast. A long or a short position will be chosen according to the forecasted value of the return; that is, if the forecasted value for a stock return is positive, a long position is chosen, otherwise a short position is chosen

4.2. Evolutionary Algorithms to Select Stocks

It is common that practitioners use indicators from the so-called fundamental analysis to assess the financial health of stocks. Besides these indicators, here, we use the stock prices and tendencies forecasted by the ANN to define which stocks should be further considered for investment. To ponder all these values, we establish an optimization problem following the recommendations in ([78]) and use an evolutionary algorithm to address it as recommended in ([10]).

Let $S = \{s_1, s_2, \ldots, card(S)\}$ be the set of considered stocks, $v_j(s_i)$ be the evaluation of stock s_i on the jth indicator, $j = 1, \ldots, N$ (for the sake of simplicity, assume that v_1 is the forecasted return as calculated by Equation (1)), and w_i be the relative importance of each indicator and forecasted return (the latter is denoted by w_1). The score of stock s_i can be calculated as follows (cf. [10,79]):

$$score(s_i) = \sum_{J=1}^{N} w_j v_j(s_i) \tag{2}$$

Since increasing $v_j(s_i)$ for $j = 1, \ldots, N$ indicates the convenience of the stock, determining the most appropriate values for w_j becomes crucial to determining the most plausible stocks as those that maximize Equation (2).

If we want to take advantage of market downtrends, sometimes we will be interested in obtaining the more negative returns to invest in a short position. To implement this idea, the value of each factor v_j is taken as positive or negative according to the prediction given by the ANN model on the previous stage. Namely, if the ANN model predicts a positive stock return, a long position will be chosen for this stock and the factor values are taken as they are. However, if the ANN model predicts a negative stock return, a short position is chosen for this stock and the return and each factor value are multiplied by -1, so Equation (2) is still valid.

To determine the most convenient values for w_j ($j = 1, \ldots, N$), we use the function recommended in ([78]). Let us define this function.

For a given historical period t, a set of predefined weights will allow one to determine the score of each stock; thus, the top, say, 5% of the stocks can be selected. These top stocks constitute the set of "selected" stocks, and the rest constitute the set of "non-selected" stocks for period t. Let $R^t_{selected}$ and $R^t_{non-selected}$ be the average returns of the stocks in these sets (as calculated by Equation (1)), respectively. The convenience of the predefined weights is then calculated as the arithmetic difference between the average returns of the selected and non-selected stocks that they produce, that is:

$$\text{Maximize } \xi(W) = \frac{1}{T} \sum_{t=1}^{T} (R^t_{selected} - R^t_{non-selected}) \tag{3}$$

where T is the number of historical returns used to assess the weights in W, $W = [w_1, w_2, \ldots, w_N]^\top$ and $\xi(W)$ is the convenience of the weights in W.

As was stated in Section 2.4, the differential evolution (DE) algorithm has been found to be highly effective in non-linear mono-objective optimization problems, especially in problems related to financial problems ([11,23,24]); therefore, this type of algorithm represents serious advantages over other optimization algorithms, particularly over other meta-heuristics. We use here a basic version of the DE algorithm as presented by Algoritm 1 in ([80]). Let us describe this algorithm.

To determine the best values for w_j $(i = 1, \ldots, N)$, the decision variables considered by the DE will be the values w_j such that each individual in the DE will contain the values for w_j fulfilling the constraints of the problem: $w_j \geq 0$ and $\sum_{j=1}^{N} w_j = 1$.

Lines 1–8 of Algorithm 1 randomly initialize the population of the DE; that is, the lines initialize feasible individuals by placing them in a random position within the search space. To ensure feasibility, the values for w_j in each individual are normalized in Lines 5 and 6.

The parameters used by the DE algorithm consist of a crossover probability, $CR \in [0, 1]$, a differential weight, $F \in [0, 2]$, and a number of individuals in the population, $population_{size} \geq 4$. Each individual in the population is represented by a real-valued vector $z = [z_1, z_2, \ldots, z_N]^\top$, where z_j is the value assigned to the jth decision variable and N is the number of decision variables (in Problem (2), the decision variables are the N weights). The termination criterion used here for the search procedure consists of a predefined number of iterations (generations). The evolutionary process is performed in Lines 9–22. Here, for each generation of the DE, the solutions in the population are evolved such that the new population is composed of the best solutions found so far. Finally, the best solution found overall is selected in Line 23.

Algorithm 1 Differential evolution used to address Problem (3).

```
Require: $N_{iterations}, CR, F, population_{size}$
Ensure: The values $w_1, w_2, \ldots, w_N$ found that best solves Problem (3)
 1: $P \leftarrow \emptyset$
 2: $i \leftarrow 1$
 3: while $(i \leq population_{size})$ do
 4:     Randomly, define $z_k \in [0, 1]$ for $z = [z_1, z_2, \ldots, z_N]^\top$
 5:     $sum \leftarrow \sum_{k=1}^{N} z_k$
 6:     $z_k \leftarrow z_k / sum$ $(k = 1, 2, \ldots, N)$
 7:     $P \leftarrow P \cup \{z\}$
 8: end while
 9: $j \leftarrow 1$
10: while $(j \leq N_{iterations})$ do
11:     for all $(z \in P)$ do
12:         Randomly, define $a, b, c \in P$, such that $z$, $a$, $b$, $c$ are all different
13:         Randomly, define $r \in \{1, \ldots, N\}$
14:         for all $(i \in \{1, \ldots, N\})$ do
15:             Randomly, define $u \in [0, 1]$
16:             If $u < CR$ or $i = r$, set $y_i = a_i + F \cdot (b_i - c_i)$, otherwise set $y_i = z_i$
17:         end for
18:         $sum \leftarrow \sum_{k=1}^{N} y_k$
19:         $y_k \leftarrow y_k / sum$ $(k = 1, 2, \ldots, N)$
20:         If $\xi(z) \leq \xi(y)$, then replace $z$ for $y$ in $P$ (see Equation (3))
21:     end for
22: end while
23: Select the individual $z \in P$ with the highest value $\xi(z)$; this individual represents the
    best set of weights $w_1, w_2, \ldots, w_N$ for Equation (2).
```

Different fundamental indicators could be more convenient for companies with different types of activities (see, e.g., [14]). We use here some fundamental indicators that can be used for trans-business companies following the works in ([10,11,13,14]). In this work, we use $N = 13$ factors to define the score of each stock as described below.

Forecasted return: Output of the ANN.
Return on equity: Net income over average shareholder's equity.
Return on asset: Net income over total assets.
Operating income margin: Operating earnings over revenue.
Net income margin: Total liabilities over total shareholder's equity.
Levered free cash flow: Amount of money the company left over after paying its financial debts.
Current ratio: Current assets over current liabilities.
Quick ratio: (Cash and equivalents + marketable securities + accounts receivable) over current liabilities.
Inventory turnover ratio: Net sales over ending inventory.
Receivable turnover ratio: Net credit sales over average accounts receivable.
Operating income growth rate: (Operating income in the current quarter − operating income at the previous quarter) over operating income in the previous quarter.
Net income growth rate: (Net income after tax in the current quarter − net income after tax at the previous quarter) over net income after tax in the previous quarter.

4.3. Optimizing Stock Portfolios

The final activity to perform stock investments consists of determining how the resources should be allocated. A given distribution of resources among the selected stocks is known as the stock portfolio. Defining the most convenient distribution of resources is known as portfolio optimization. In this final activity, the decision alternatives are no longer individual stocks but complete portfolios. Thus, it is necessary to determine multiple criteria to comprehensively assess portfolios.

Formally, a stock portfolio is a vector $x = [x_1, x_2, \ldots, x_m]^\top$ such that x_i is the proportion of the total investment that is allocated to the ith stock. Let r_i be the return of the ith stock calculated according to Equation (1); the return of a given portfolio x is defined as follows:

$$R(x) = \sum_{i=1}^{m} x_i r_i \tag{4}$$

Of course, if we knew the $t + 1$ return of the stocks, we could allocate resources that maximize $R(x)$ without uncertainty; however, since this is impossible, the multiple criteria used to assess portfolios are estimations of $R(x)$. These estimations usually come from probability theory.

According to ([12]), the most convenient portfolio x can be determined by optimizing a set of confidence intervals that describe the probabilistic distribution of the portfolio's return:

$$\underset{x \in \Omega}{\text{Maximize}} \{\theta(x) = (\theta_{\beta_1}(x), \theta_{\beta_2}(x), \ldots, \theta_{\beta_k}(x))\} \tag{5}$$

where $\theta_{\beta_i}(x) = \{[c_i, d_i] : P(c_i \leq E(R(x)) \leq d_i) = \beta_i\}$, $E(R(x))$ is the expected return of portfolio x, $P(\omega)$ is the probability that event ω occurs and Ω is the set of feasible portfolios.

Maximizing confidence intervals as conducted in Equation (5) does not mean increasing the wideness of the intervals; rather, it refers to the intuition that rightmost returns in the probability distribution are desired. We use the so-called interval theory ([68]) to measure the possibility that a confidence interval is greater than another one. In interval theory, an interval number allows one to encompass the uncertainty involved in the definition of a quantity.

Since we are trying to find the best portfolios in terms of confidence intervals around their expected return, intervals further to the right are better (rather than comparing intervals in terms of their width). Therefore, the comparison method used must provide this feature. There are several works in the literature describing methods that possess this property (e.g., [81,82]); however, the method proposed in [83] is the most broadly mentioned in the literature [84].

The authors of [83] presented a possibility function to define the order between two interval numbers that has been increasingly used in the literature (e.g., [12,85–87]). Let $I = [i^-, i^+]$ and $J = [j^-, j^+]$ be two interval numbers, and the possibility function presented in [83] is defined as follows:

$$possibility(I \geq J) := \begin{cases} 1, & \text{if } p(I,J) > 1 \\ 0, & \text{if } p(I,J) < 0 \\ p(I,J), & \text{otherwise} \end{cases}$$

where $p(I,J) = \frac{i^+ - j^-}{(i^+ - i^-) + (j^+ - j^-)}$.

Moreover, if $i = i^+ = j^-$ and $j = j^+ = j^-$, then

$$possibility(I \geq J) := \begin{cases} 1, & \text{if } i \geq j \\ 0, & \text{otherwise} \end{cases}$$

Since Problem (5) can potentially have many objectives defined as interval numbers as well as multiple constraints, we use MOEA/D (see Section 2.4), as advised by ([12]). In ([12]), MOEA/D was adapted to deal with these types of objectives; the adaptation has been proven to provide good results in contexts related to stock investments. For reasons of space in this paper, the reader is referred to ([12]) for specific details about this improvement to MOEA/D.

5. Experiments

The hypothesis that a procedure that comprehensively addresses the practitioners' main activities while also considering uptrends and downtrends produce better total earnings for the investor than when not doing it is tested by using extensive experiments with actual historical data.

5.1. Experimental Design

We used well-known data for our experiments; the historical prices and financial information about the stocks within the Standard and Poor's 500 (S&P500) index. The officially reported financial information was used to build criterion performances.

Data from some of the most recent ninety months were used as input in the experiments, i.e., from November 2013 to April 2021. This dataset contains both uptrends and downtrends, so it is convenient for the kind of tests performed here. From these periods, sixty are used to prepare (say, train) the algorithms, and the rest are used to assess the approach performance in a window-sliding manner. For example, the information on November 2013–October 2018 is used to determine the investments that should be carried out at the beginning of November 2018, and these investments are maintained the whole month. Then, the performance of the approach (i.e., the returns) is calculated at the end of November 2018 using Equation (4). Such a performance is compared to the benchmarks in that period. Later, the investments are neglected, and, independently, the lapse is slid one period; thus, now, the information of the sixty months—December 2013–November 2018—is used to determine the investments for December 2018, where the new approach performance is calculated and compared to the benchmarks. This procedure is repeated thirty times; so, the conclusions can shed light on the robustness and overall performance of the approach with a high degree of confidence.

5.2. Benchmarks

The Standard and Poor's 500 index is used to define the relative performance of the proposed approach. Stock indexes are often used by practitioners as benchmarks because they summarize valuable information regarding the main sectors of an economy. The S&P500 is perhaps the most well-known and used index; it aggregates information about the five hundred biggest publicly traded companies in the United States of America. Since we are making decisions considering information only from this index, comparing the performance of the proposed approach with it is fair. In addition to the S&P500 index, in order to validate our approach, we have included several benchmarks to measure the effectiveness of our proposal. These benchmarks are: the approach of ([10]), the approach of ([12]) and our approach without including downtrends.

5.3. Parameter Setting

The parameter values used by each of the techniques mentioned in Section 4 are defined here.

As explained above, the number of periods used to train the ANN for each stock is 60. The only hidden layer uses sixteen neurons. We observed in preliminary experiments that the ANN showed more efficiency when it uses the same number of neurons as inputs; the more neurons, the more unstable the ANN was and the fewer neurons, the less predictive capacity the ANN had. Each neuron of the ANN used the sigmoidal function as the activation function. The ANN was run $n_a = 50$ times to train the ANN for each stock; finally, the ANN model with fewer testing errors was used to predict the return at time $t + 1$.

Regarding the selection of stocks, the DE defined to select the factor weights that maximize the objective function shown in Equation (3) uses common parameter values. The crossover probability was set to 0.9; the differential weight was set to 0.8; the population size was set to 200; the number of iterations was set to 100. After scoring and ranking the stocks, we only select the top 5% of all the stocks originally considered following the recommendations in [10].

Finally, the genetic algorithm used to address Problem (5) was described in detail in ([12]), where it was based on the well-known MOEA/D and adapted to deal with parameter values defined as interval numbers. We use one hundred generations as the stopping criterion, two solutions as the maximum number of solutions replaced by each child solution, a probability of selecting parents only from the neighborhood (instead of the whole population) of 0.9, one hundred subproblems, and twenty weight vectors in the neighborhood of each weight vector. Two confidence intervals are considered by MOEA/D as objectives to be maximized (see Equation (5)): $\theta_{\beta_{30}}(x)$ and $\theta_{\beta_{50}}(x)$ according to the recommendations in ([12]). The constraints considered by MOEA/D are $x_i \geq 0$ and $\sum x_i = 1$.

It is worth mentioning that the code for implementing the algorithms described here are original developments of the authors. The code was written in Matlab and Java and will be probably publicly presented in the form of a complete software system.

5.4. Results

The proposed approach uses components that exploit randomness to explore the search space. Here, we intend to discard the effects produced by such randomness by running our approach many times; particularly, each stochastic component runs twenty times for each of the thirty back-testing periods mentioned in Section 5.1. Doing it this way sheds light on the robustness of our approach and allows us to reach sound conclusions. By following the recommendations in [88], the performance of our approach is evaluated by using the quantiles Q_{10}, Q_{20}, Q_{50} (median), Q_{80} and Q_{90} (see Figure 1). As is noted in [88], distribution solutions of stochastic optimization algorithms are often asymmetrical; hence by using quantiles, we could obtain more insights into our approaches. However, Figure 1 shows that, in our case, Q_{50} and the mean are almost always overlapped. Furthermore, (Q_{10}, Q_{20}) and (Q_{80}, Q_{90}) are symmetrical with respect to the mean. This behavior indicate that the performance of our approach is practically normally distributed.

Therefore, in this study, the average returns of our approach is used to be compared with several benchmarks, as shown in Table 1 and Figure 2. For simplicity, the results are discussed hereafter as if the returns were not averages. In order to validate our approach, in this section, we have included several benchmarks to measure the effectiveness of our proposal. These benchmarks are: (a) the market index S&P500, (b) the approach of [10], (c) the approach of [12] and (d) our approach without including downtrends.

From Table 1 and Figure 2, we can see that, in terms of the expected value, the worst overall return was produced by investing according to the S&P500 index, while the best overall return was achieved by investing in a portfolio produced by the proposed approach that takes advantage of both positive and negative trends. Figure 2 shows that the portfolio that considers negative trends is almost always in the top two from all the approaches. Furthermore, the returns obtained using this approach show that this model is not affected by the downtrends in the market as the benchmarks, as seen in the fall of all approaches from Jan 2020 to Mar 2020. Remarkably, this behavior did not prevent the proposed approach from exploiting the clear overall uptrend produced from Apr. 2020 to Apr. 2021, as can be clearly seen in Table 1.

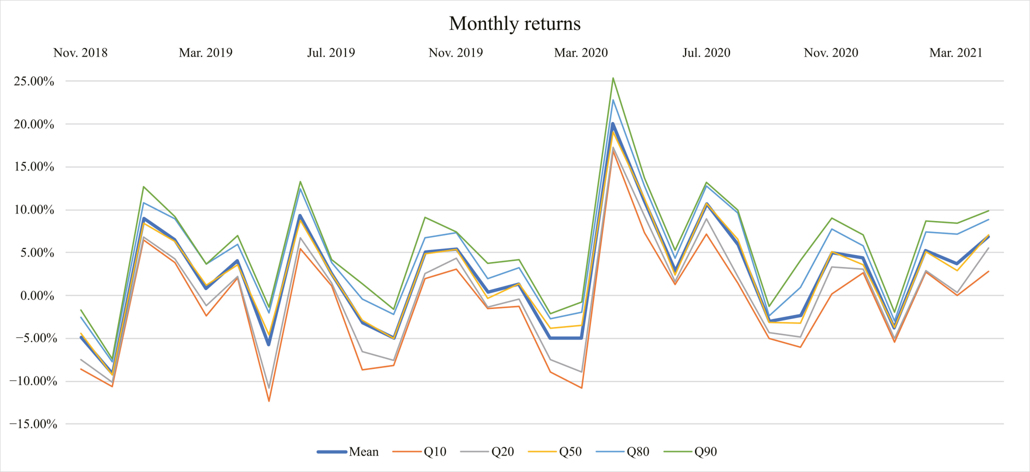


Figure 1. Monthly returns of our approach for the mean and several quantiles.

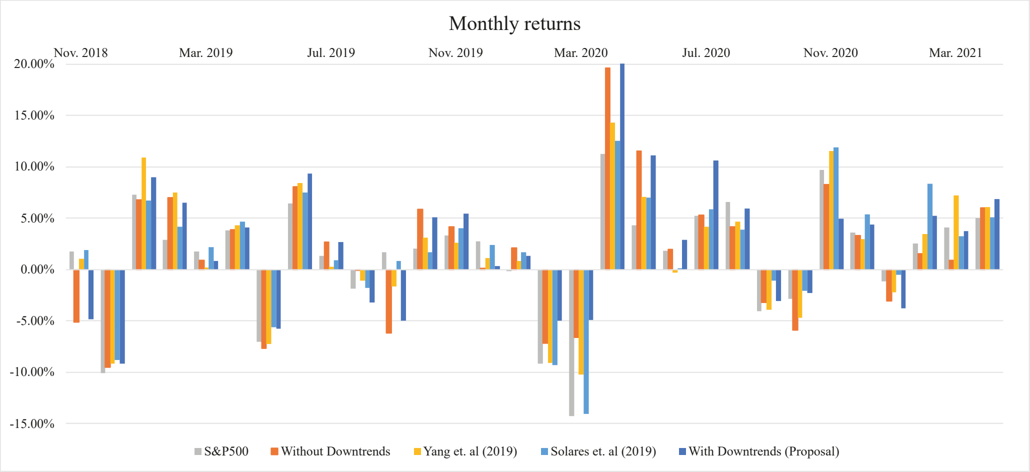


Figure 2. Monthly returns comparison.

Table 1. Returns produced per period. In the case of the algorithms, the return is averaged in twenty runs.

	S&P500 Index	**Yang et al. (2019)**	**Solares et al. (2019)**	**Without Negative Trends**	**With Negative Trends**
Nov. 2018	1.75%	1.01%	1.87%	−5.11%	−4.87%
Dec. 2018	−10.11%	−9.18%	−8.81%	−9.56%	−9.14%
Jan. 2019	7.29%	10.88%	6.71%	6.77%	9.00%
Feb. 2019	2.89%	7.47%	4.19%	7.00%	6.52%
Mar. 2019	1.76%	0.20%	2.17%	0.89%	0.81%
Apr. 2019	3.78%	4.29%	4.65%	3.88%	4.06%
May. 2019	−7.04%	−7.22%	−5.65%	−7.66%	−5.77%
Jun. 2019	6.45%	8.45%	7.53%	8.06%	9.33%
Jul. 2019	1.30%	0.25%	0.92%	2.66%	2.64%
Aug. 2019	−1.84%	−1.08%	−1.78%	−0.03%	−3.19%
Sep. 2019	1.69%	−1.63%	0.83%	−6.20%	−4.96%
Oct. 2019	2.00%	3.12%	1.67%	5.85%	5.09%
Nov. 2019	3.29%	2.58%	4.00%	4.17%	5.43%
Dec. 2019	2.78%	1.13%	2.39%	0.13%	0.36%
Jan. 2020	−0.16%	0.81%	1.67%	2.13%	1.29%
Feb. 2020	−9.18%	−9.09%	−9.28%	−7.22%	−4.96%
Mar. 2020	−14.30%	−10.27%	−14.03%	−6.59%	−4.94%
Apr. 2020	11.26%	14.33%	12.53%	19.64%	20.02%
May. 2020	4.33%	7.09%	7.02%	11.54%	11.11%
Jun. 2020	1.81%	−0.29%	0.15%	1.95%	2.88%
Jul. 2020	5.22%	4.18%	5.87%	5.28%	10.65%
Aug. 2020	6.55%	4.68%	3.90%	4.18%	5.94%
Sep. 2020	−4.08%	−3.95%	−1.10%	−3.20%	−3.04%
Oct. 2020	−2.85%	−4.74%	−2.05%	−5.88%	−2.31%
Nov. 2020	9.71%	11.50%	11.91%	8.28%	4.97%
Dec. 2020	3.58%	2.95%	5.39%	3.33%	4.37%
Jan. 2021	−1.13%	−2.23%	−0.53%	−3.06%	−3.76%
Feb. 2021	2.54%	3.43%	8.35%	1.51%	5.23%
Mar. 2021	4.07%	7.22%	3.23%	0.88%	3.75%
Apr. 2021	4.98%	6.05%	5.09%	6.00%	6.84%
Average	**1.28%**	**1.73%**	**1.96%**	**1.65%**	**2.45%**
Std desv.	**5.61%**	**6.06%**	**5.76%**	**6.35%**	**6.27%**

From Table 2, we can see that the proposed approach outperforms the benchmarks at the end of the thirty periods: the sum of returns is approximately 41% better than Yang et al. 2019 ([10]), 25% better than Solares et al. 2019 ([12]), 48% better than the one that only considers positive trends and more than 90% better than the market index. Moreover, the cumulative returns of our proposal is 63% better than Yang et al. 2019 ([10]), 35% better than Solares et al. 2019 ([12]), 75% better than the one that only considers positive trends and 141% better than the market index. This performance can be seen in Figures 3 and 4.

Both Figures 3 and 4 describe the evolution of the portfolio returns in an aggregate way throughout the whole time lapse (i.e., November 2018 to April 2021). However, Figure 3 shows this evolution from the perspective of the sums of the returns, while Figure 4 shows the cumulative returns. Both figures can be relevant to the practitioner. The former shows the overall performance of the approach without considering the exact period where the return was obtained, while Figure 4 allows one to ponder the impact of the period where such a return was obtained. Let us unfold the latter. Figure 4 shows the amount that the investor would obtain if he/she takes their investment in a given period. For instance, an investment of USD 1000 at the beginning of November 2018 using the proposed model would have become USD 992 (i.e., −0.80%) if the investor would have withdrawn the investment at the end of May 2019. However, if he/she continues until April 2021, the investment would have become USD 1952 (i.e., +95.28%). In this sense, it is clear that the proposed approach outperformed the benchmarks by creating a portfolio that includes long and short positions. This result shows the potential of our proposal, which could be improved in future approaches by including stocks from other indexes, more technical/fundamental variables, etc.

In both Figures, it can be seen that, for the first fourteen periods (November 2018 to December 2019), the market does not move significantly in any direction; however, for the remaining periods, the market starts both negative and positive trends. A higher final return achieved by our proposal indicates that it is taking advantage of these trends overall. These results also show that considering negative trends is crucial. The figures show that the final average return is better if negative trends are considered when building stock portfolios.

Table 2. Sum of returns and cumulative returns. In the case of the algorithms, the return is averaged over twenty runs.

	Sum of Returns					**Cumulative Returns**				
	S&P500 Index	**Yang et al. (2019)**	**Solares et al. (2019)**	**Without Down-Trends**	**With Down-Trends**	**S&P500 Index**	**Yang et al. (2019)**	**Solares et al. (2019)**	**Without Down-Trends**	**With Down-Trends**
Nov. 2018	1.75%	1.01%	1.87%	−5.11%	−4.87%	1.75%	1.01%	1.87%	−5.11%	−4.87%
Dec. 2018	−8.35%	−8.17%	−6.94%	−14.67%	−14.01%	−8.53%	−8.27%	−7.11%	−14.18%	−13.56%
Jan. 2019	−1.06%	2.70%	−0.24%	−7.89%	−5.01%	−1.86%	1.71%	−0.88%	−8.37%	−5.79%
Feb. 2019	1.83%	10.17%	3.95%	−0.89%	1.51%	0.98%	9.31%	3.28%	−1.95%	0.36%
Mar. 2019	3.59%	10.37%	6.12%	0.00%	2.32%	2.76%	9.53%	5.52%	−1.08%	1.17%
Apr. 2019	7.37%	14.66%	10.77%	3.88%	6.38%	6.64%	14.23%	10.42%	2.76%	5.28%
May. 2019	0.33%	7.44%	5.12%	−3.78%	0.61%	−0.87%	5.98%	4.18%	−5.11%	−0.80%
Jun. 2019	6.78%	15.89%	12.65%	4.28%	9.94%	5.53%	14.93%	12.03%	2.54%	8.46%
Jul. 2019	8.08%	16.14%	13.58%	6.94%	12.58%	6.89%	15.22%	13.07%	5.27%	11.32%
Aug. 2019	6.24%	15.06%	11.80%	6.91%	9.39%	4.93%	13.97%	11.05%	5.23%	7.77%
Sep. 2019	7.92%	13.43%	12.63%	0.70%	4.43%	6.70%	12.11%	11.98%	−1.29%	2.43%
Oct. 2019	9.93%	16.54%	14.30%	6.56%	9.52%	8.83%	15.61%	13.85%	4.48%	7.64%
Nov. 2019	13.22%	19.12%	18.30%	10.72%	14.95%	12.42%	18.59%	18.40%	8.84%	13.48%
Dec. 2019	16.00%	20.25%	20.68%	10.85%	15.31%	15.54%	19.93%	21.22%	8.97%	13.89%
Jan. 2020	15.84%	21.06%	22.36%	12.98%	16.60%	15.35%	20.90%	23.25%	11.30%	15.36%
Feb. 2020	6.65%	11.98%	13.07%	5.76%	11.64%	4.76%	9.92%	11.81%	3.26%	9.64%
Mar. 2020	−7.65%	1.71%	−0.96%	−0.83%	6.70%	−10.22%	−1.37%	−3.88%	−3.54%	4.23%
Apr. 2020	3.61%	16.04%	11.57%	18.81%	26.73%	−0.12%	12.77%	8.16%	15.40%	25.10%
May. 2020	7.94%	23.13%	18.58%	30.36%	37.84%	4.21%	20.76%	15.75%	28.72%	38.99%
Jun. 2020	9.75%	22.84%	18.73%	32.31%	40.72%	6.09%	20.41%	15.91%	31.24%	43.00%
Jul. 2020	14.97%	27.01%	24.60%	37.59%	51.37%	11.63%	25.43%	22.72%	38.16%	58.23%
Aug. 2020	21.52%	31.69%	28.51%	41.77%	57.31%	18.94%	31.30%	27.51%	43.94%	67.63%
Sep. 2020	17.43%	27.74%	27.41%	38.57%	54.28%	14.09%	26.12%	26.11%	39.33%	62.54%
Oct. 2020	14.59%	23.01%	25.36%	32.68%	51.97%	10.84%	20.14%	23.53%	31.13%	58.80%
Nov. 2020	24.30%	34.51%	37.27%	40.97%	56.94%	21.60%	33.97%	38.24%	41.99%	66.69%
Dec. 2020	27.88%	37.47%	42.66%	44.30%	61.31%	25.96%	37.92%	45.70%	46.73%	73.97%
Jan. 2021	26.75%	35.23%	42.14%	41.24%	57.55%	24.54%	34.85%	44.93%	42.24%	67.43%
Feb. 2021	29.29%	38.66%	50.48%	42.75%	62.78%	27.70%	39.47%	57.03%	44.39%	76.18%
Mar. 2021	33.36%	45.88%	53.71%	43.63%	66.52%	32.90%	49.54%	62.10%	45.66%	82.78%
Apr. 2021	38.35%	51.93%	58.80%	49.64%	73.36%	39.52%	58.59%	70.35%	54.41%	95.28%

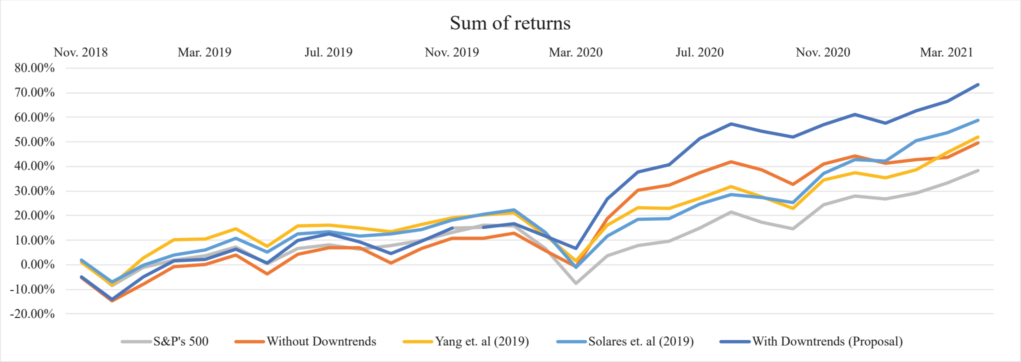


Figure 3. Sum of returns comparison.

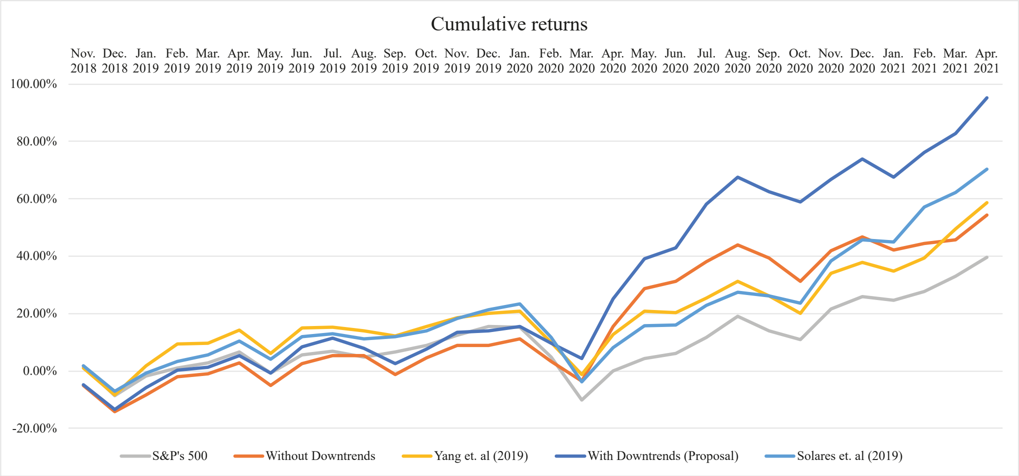


Figure 4. Cumulative returns comparison.

According to Figure 2 and Table 1, the worst return obtained by our approach was in Dec 2018. This is also shown in Figure 4, where the detriment is caused by the higher negative return produced in this period. In that moment, the system decided to open long positions and allocate high proportions of investments to some stocks with bad actual returns. This was due to the good historical performance of such actions that indicated a good statistical behavior. Several external issues affect the performance of a stock in the market, such as the case of Nvidia corporation as reported in the news [89]. Thus, a way of improving the proposed system in the future is by considering criteria coming from the so-called sentiment analysis [90] that takes into consideration such factors.

On the other hand, as a way of measuring the performance of our proposal and comparing the results with some benchmarks, the Sharpe ratio r_{sharpe} and Sortino ratio $r_{sortino}$ are used. These ratios are defined as

$$r_{sharpe} = \frac{R_p - R_f}{\sigma_p}$$

and

$$r_{sortino} = \frac{R_p - R_f}{\sigma_{pd}}$$

where R_p is the average portfolio return, R_f is the best available risk-free security rate, σ_p is the portfolio standard deviation and $\sigma_{p,d}$ is the portfolio standard deviation of the downside. These indexes measure the risk per return obtained in comparison with a risk-free asset. In particular, the Sharpe ratio describes how much return is received per unit of risk; meanwhile, the Sortino ratio describes how much return is received per unit of **bad** risk. Therefore, the higher these indexes are, the more convenient for investment the asset is. We have considered the Treasure Bond of USA a risk-free security, with a value of 3% of annual return. We also considered the Treasure Bond of USA as the minimal acceptance ratio (MAR) to compute the downside deviation. R_p, σ_p and $\sigma_{p,d}$ are taken from Table 1. Table 3 shows the Sharpe and Sortino ratios for the all the benchmarks and our proposal. According to the results, our proposal has the best performance for both indexes; overall, it has higher returns by considering the risk.

Table 3. Comparison of benchmarks with the proposal by using the Sharpe and Sortino ratios.

	Sharpe Ratio	**Sortino Ratio**
S&P's 500	0.1831	0.2529
Yang et al. (2019)	0.2445	0.4072
Solares et al. (2019)	0.2966	0.4551
Without downtrends	0.2213	0.3884
With downtrends	0.3502	0.7223

6. Conclusions

Building stock portfolios with high returns and low risk is a common challenge for researchers in the financial area. Usually, the most common practice is to select the more promising stocks according to several factors, such as financial information, news of the market and technical analysis. Several approaches that use computational intelligence algorithms have been proposed in the literature to deal with the overwhelming complexity of building a stock portfolio. Usually, these approaches consider up to three activities to build a portfolio: return forecasting, stock selection and portfolio optimization. These activities decide which stocks should be supported, as well as the proportions of the investment to be allocated to them, by comparing the historical and forecasted performance of potential stock investments. However, to the best of our knowledge, these approaches do not comprehensively address the three activities when considering downtrends in stock prices.

In this paper, a comprehensive approach to stock portfolio management is proposed; the approach includes stock price forecasting, stock selection and stock portfolio optimization while taking advantage of market downtrends.

Stock price forecasting is carried out through an artificial neural network (ANN) trained by the extreme learning machine (ELM) algorithm. Forecasting the price of a given stock allows the comprehensive approach to focus on uptrends or downtrends (i.e., going long or short, respectively) for that stock. Stock selection is modeled as an optimization problem that seeks to determine the most plausible stocks; thus, a differential evolution is exploited on the basis of the forecasted price and a set of factors of the so-called fundamental analysis. Finally, portfolio optimization is conducted through a genetic algorithm that uses confidence intervals of the portfolio returns to determine the best stock portfolio.

Using preliminary experimentation, we found that the ELM was better than other methods (ANN with back-propagation, random forest, support vector regression) at forecasting the trend of the stock price but not the best at forecasting stock returns. Therefore, more research should be conducted to discover better configurations of the ANN with ELM or to decide if the forecasting stage should be changed. However, further research on this, as well as on methods to increase the performance of the next stages of the comprehensive approach, is beyond the scope of this work, so the authors will address these issues in future works.

Regarding the assessment of the comprehensive approach, the obtained results show that stock selection and portfolio optimization stages make more profitable portfolios when negative trends of stocks are taken into account to take advantage of downtrends of the market (see Table 2 and Figures 3 and 4). Furthermore, the results show that not only a traditional benchmark, the Standard and Poor's 500 index, is outperformed by the proposed approach but also approaches that do not exploit negative market trends (e.g., [10,12]).

This research work could be improved by the following possible future directions:

I A deeper study of the forecasting stage to test the performance of several AI methods by employing more data or different financial variables;

II A deeper study on the selection stage to evaluate the performance of the system by employing different financial variables to build the stock portfolio;

III A deeper study of the performance of the system by modifying different parameters in the optimization stage and comparing the results with other approaches;

IV New experiments to show the robustness of the approach regarding (i) the number and type of alternatives in the universe of stocks, (ii) the number of selected stocks and (iii) the parameter values.

Author Contributions: Conceptualization, E.S. and V.d.-L.-G.; methodology, E.S.; software, E.S., F.G.S. and V.d.-L.-G.; validation, F.G.S., E.S. and V.d.-L.-G.; formal analysis, F.G.S. and R.D.; investigation, R.D.; resources, R.D.; data curation, V.d.-L.-G. and F.G.S.; writing—original draft preparation, E.S.; writing—review and editing, F.G.S. and V.d.-L.-G.; visualization, R.D. and V.d.-L.-G.; supervision, R.D. and F.G.S.; funding acquisition, R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Instituto Tecnológico y de Estudios Superiores de Monterrey, and by the Mexican National Council of Science and Technology (CONACYT) grant number 321028, and by SEP-PRODEP México grant numbers UACOAH-PTC-545 and UACOAH-CA-479. The APC was funded by Instituto Tecnológico y de Estudios Superiores de Monterrey.

Acknowledgments: The work of Raymundo Díaz was supported by the vice president of Research of Tecnológico de Monterrey. Efrain Solares thanks the Mexican National Council of Science and Technology (CONACYT) for its support to project no. 321028 and SEP-PRODEP México for its support under grant UACOAH-PTC-545. Francisco G. Salas and Víctor De-Leon-Gomez were supported by SEP-PRODEP México.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pan, H.; Long, M. Intelligent Portfolio Theory and Application in Stock Investment with Multi-Factor Models and Trend Following Trading Strategies. *Procedia Comput. Sci.* **2021**, *187*, 414–419. [CrossRef]
2. Jiang, X.; Peterburgsky, S. Investment performance of shorted leveraged ETF pairs. *Appl. Econ.* **2017**, *49*, 4410–4427. [CrossRef]
3. Hurlin, C.; Iseli, G.; Pérignon, C.; Yeung, S. The counterparty risk exposure of ETF investors. *J. Bank. Financ.* **2019**, *102*, 215–230. [CrossRef]
4. Holzhauer, H.M.; Lu, X.; McLeod, R.W.; Mehran, J. Bad news bears: Effects of expected market volatility on daily tracking error of leveraged bull and bear ETFs. *Manag. Financ.* **2013**, *39*, 1169–1187.
5. Gregory-Allen, R.B.; Smith, D.M.; Werman, M. Chapter 30—Short Selling by Portfolio Managers: Performance and Risk Effects across Investment Styles. In *Handbook of Short Selling*; Gregoriou, G.N., Ed.; Academic Press: San Diego, CA, USA, 2012; pp. 437–451. [CrossRef]
6. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
7. Li, X.; Xie, H.; Wang, R.; Cai, Y.; Cao, J.; Wang, F.; Min, H.; Deng, X. Empirical analysis: Stock market prediction via extreme learning machine. *Neural Comput. Appl.* **2016**, *27*, 67–78. [CrossRef]
8. Wang, F.; Zhang, Y.; Rao, Q.; Li, K.; Zhang, H. Exploring mutual information-based sentimental analysis with kernel-based extreme learning machine for stock prediction. *Soft Comput.* **2017**, *21*, 3193–3205. [CrossRef]
9. Das, S.P.; Padhy, S. Unsupervised extreme learning machine and support vector regression hybrid model for predicting energy commodity futures index. *Memetic Comput.* **2017**, *9*, 333–346. [CrossRef]
10. Yang, F.; Chen, Z.; Li, J.; Tang, L. A novel hybrid stock selection method with stock prediction. *Appl. Soft Comput.* **2019**, *80*, 820–831. [CrossRef]
11. Fernandez, E.; Navarro, J.; Solares, E.; Coello, C.C. A novel approach to select the best portfolio considering the preferences of the decision maker. *Swarm Evol. Comput.* **2019**, *46*, 140–153. [CrossRef]
12. Solares, E.; Coello, C.A.C.; Fernandez, E.; Navarro, J. Handling uncertainty through confidence intervals in portfolio optimization. *Swarm Evol. Comput.* **2019**, *44*, 774–787. [CrossRef]
13. Xidonas, P.; Mavrotas, G.; Psarras, J. A multicriteria methodology for equity selection using financial analysis. *Comput. Oper. Res.* **2009**, *36*, 3187–3203. [CrossRef]
14. Marasović, B.; Poklepović, T.; Aljinović, Z. MArkowitz'model with fundamental and technical analysis–complementary methods or not. *Croat. Oper. Res. Rev.* **2011**, *2*, 122–132.
15. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [CrossRef] [PubMed]
16. Sumathi, S.; Paneerselvam, S. *Computational Intelligence Paradigms: Theory & Applications Using MATLAB*; CRC Press: New York, NY, USA, 2010.

17. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
18. Coello, C.A.C.; Lamont, G.B.; Van Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-Objective Problems*; Springer: Berlin, Germany, 2007; Volume 5.
19. Bianchi, L.; Dorigo, M.; Gambardella, L.M.; Gutjahr, W.J. A survey on metaheuristics for stochastic combinatorial optimization. *Nat. Comput.* **2009**, *8*, 239–287. [CrossRef]
20. Pławiak, P. Novel genetic ensembles of classifiers applied to myocardium dysfunction recognition based on ECG signals. *Swarm Evol. Comput.* **2018**, *39*, 192–208. [CrossRef]
21. Pławiak, P. Novel methodology of cardiac health recognition based on ECG signals and evolutionary-neural system. *Expert Syst. Appl.* **2018**, *92*, 334–349. [CrossRef]
22. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2010**, *15*, 4–31. [CrossRef]
23. Krink, T.; Paterlini, S. Multiobjective optimization using differential evolution for real-world portfolio optimization. *Comput. Manag. Sci.* **2011**, *8*, 157–179. [CrossRef]
24. Krink, T.; Mittnik, S.; Paterlini, S. Differential evolution and combinatorial search for constrained index-tracking. *Ann. Oper. Res.* **2009**, *172*, 153. [CrossRef]
25. Zhang, Q.; Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
26. Sharma, D.K.; Hota, H.; Brown, K.; Handa, R. Integration of genetic algorithm with artificial neural network for stock market forecasting. *Int. J. Syst. Assur. Eng. Manag.* **2021**, 1–14. [CrossRef]
27. Ferreira, F.; Gandomi, A.H.; Cardoso, R.T.N. Artificial Intelligence Applied to Stock Market Trading: A Review. *IEEE Access* **2021**, *9*, 30898–30917. [CrossRef]
28. Chopra, R.; Sharma, G.D. Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda. *J. Risk Financ. Manag.* **2021**, *14*, 526. [CrossRef]
29. Ma, Y.L.; Han, R.Z.; Wang, W.Z. Prediction-Based Portfolio Optimization Models Using Deep Neural Networks. *IEEE Access* **2020**, *8*, 115393–115405. [CrossRef]
30. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [CrossRef]
31. Long, W.; Lu, Z.; Cui, L. Deep learning-based feature engineering for stock price movement prediction. *Knowl.-Based Syst.* **2019**, *164*, 163–173. [CrossRef]
32. Zhong, X.; Enke, D. Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financ. Innov.* **2019**, *5*, 1–20. [CrossRef]
33. Kaczmarek, T.; Perez, K. Building portfolios based on machine learning predictions. *Econ. Res.-Ekon. Istraz.* **2021**, 1–19. [CrossRef]
34. Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Syst. Appl.* **2015**, *42*, 259–268. [CrossRef]
35. Chong, E.; Han, C.; Park, F.C. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Syst. Appl.* **2017**, *83*, 187–205. [CrossRef]
36. Huang, G.B.; Siew, C.K. Extreme learning machine: RBF network case. In Proceedings of the 2004 8th International Conference on Control, Automation, Robotics and Vision (ICARCV) ICARCV 2004, Kunming, China, 6–9 December 2004; Volume 2, pp. 1029–1036. [CrossRef]
37. Peykani, P.; Mohammadi, E.; Jabbarzadeh, A.; Rostamy-Malkhalifeh, M.; Pishvaee, M.S. A novel two-phase robust portfolio selection and optimization approach under uncertainty: A case study of Tehran stock exchange. *PLoS ONE* **2020**, *15*, e239810. [CrossRef] [PubMed]
38. Mussafi, N.S.M.; Ismail, Z. Optimum Risk-Adjusted Islamic Stock Portfolio Using the Quadratic Programming Model: An Empirical Study in Indonesia. *J. Asian Financ. Econ. Bus.* **2021**, *8*, 839–850. [CrossRef]
39. Lim, S.; Kim, M.J.; Ahn, C.W. A Genetic Algorithm (GA) Approach to the Portfolio Design Based on Market Movements and Asset Valuations. *IEEE Access* **2020**, *8*, 140234–140249. [CrossRef]
40. Wang, W.Y.; Li, W.Z.; Zhang, N.; Liu, K.C. Portfolio formation with preselection using deep learning from long-term financial data. *Expert Syst. Appl.* **2020**, *143*, 113042. [CrossRef]
41. Zhang, C.; Liang, S.; Lyu, F.; Fang, L. Stock-index tracking optimization using auto-encoders. *Front. Phys.* **2020**, *8*, 388. [CrossRef]
42. Paiva, F.D.; Cardoso, R.T.N.; Hanaoka, G.P.; Duarte, W.M. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Syst. Appl.* **2019**, *115*, 635–655. [CrossRef]
43. Galankashi, M.R.; Rafiei, F.M.; Ghezelbash, M. Portfolio selection: A fuzzy-ANP approach. *Financ. Innov.* **2020**, *6*, 34. [CrossRef]
44. Markowitz, H. Portfolio Selection. *J. Financ.* **1952**, *7*, 77–91.
45. Kalayci, C.B.; Ertenlice, O.; Akbay, M.A. A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Syst. Appl.* **2019**, *125*, 345–368. [CrossRef]
46. Rockafellar, R.; Uryasev, S. Optimization of conditional value-at-risk. *J. Risk* **2002**, *2*, 21–41. [CrossRef]
47. Rockafellar, R.; Uryasev, S. Conditional value-at-risk for general loss distributions. *J. Bank. Financ.* **2002**, *26*, 1443–1471. [CrossRef]
48. Sehgal, R.; Mehra, A. Robust reward–risk ratio portfolio optimization. *Int. Trans. Oper. Res.* **2021**, *28*, 2169–2190. [CrossRef]

49. Hu, Y.; Lindquist, W.B.; Rachev, S.T. Portfolio Optimization Constrained by Performance Attribution. *J. Risk Financ. Manag.* **2021**, *14*, 12. [CrossRef]
50. Dai, Z.; Wen, F. Some improved sparse and stable portfolio optimization problems. *Financ. Res. Lett.* **2018**, *27*, 46–52. [CrossRef]
51. Baykasoğlu, A.; Yunusoglu, M.G.; Özsoydan, F.B. A GRASP based solution approach to solve cardinality constrained portfolio optimization problems. *Comput. Ind. Eng.* **2015**, *90*, 339–351. [CrossRef]
52. Mayambala, F.; Rönnberg, E.; Larsson, T. Eigendecomposition of the Mean-Variance Portfolio Optimization Model. In *Optimization, Control, and Applications in the Information Age*; Migdalas, A., Karakitsiou, A., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 209–232.
53. Kocadağli, O.; Keskin, R. A novel portfolio selection model based on fuzzy goal programming with different importance and priorities. *Expert Syst. Appl.* **2015**, *42*, 6898–6912. [CrossRef]
54. He, F.; Qu, R. Hybridising Local Search With Branch-And-Bound For Constrained Portfolio Selection Problems. In Proceedings of the 30th European Council for Modeling and Simulation, Regensburg, Germany, 31 May–3 June 2016; Claus, T., Herrmann, F., Manitz, M., Rose, O., Eds.; Digital Library of the European Council for Modelling and Simulation: Regensburg, Germany, 2016; pp. 1–7.
55. Ruiz-Torrubiano, R.; Suárez, A. A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs. *Appl. Soft Comput.* **2015**, *36*, 125–142. [CrossRef]
56. Soleymani, F.; Paquet, E. Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath. *Expert Syst. Appl.* **2020**, *156*, 113456. [CrossRef]
57. García, F.; Guijarro, F.; Oliver, J. Index tracking optimization with cardinality constraint: A performance comparison of genetic algorithms and tabu search heuristics. *Neural Comput. Appl.* **2018**, *30*, 2625–2641. [CrossRef]
58. Hadi, A.S.; Naggar, A.A.E.; Bary, M.N.A. New model and method for portfolios selection. *Appl. Math. Sci.* **2016**, *10*, 263–288. [CrossRef]
59. Liagkouras, K.; Metaxiotis, K. A new efficiently encoded multiobjective algorithm for the solution of the cardinality constrained portfolio optimization problem. *Ann. Oper. Res.* **2018**, *267*, 281–319. [CrossRef]
60. Macedo, L.L.; Godinho, P.; Alves, M.J. Mean-semivariance portfolio optimization with multiobjective evolutionary algorithms and technical analysis rules. *Expert Syst. Appl.* **2017**, *79*, 33–43. [CrossRef]
61. Lwin, K.T.; Qu, R.; MacCarthy, B.L. Mean-VaR portfolio optimization: A nonparametric approach. *Eur. J. Oper. Res.* **2017**, *260*, 751–766. [CrossRef]
62. Ban, G.Y.; Karoui, N.E.; Lim, A.E.B. Machine Learning and Portfolio Optimization. *Manag. Sci.* **2016**, *64*, 1136–1154. [CrossRef]
63. Kizys, R.; Juan, A.; Sawik, B.; Calvet, L. A Biased-Randomized Iterated Local Search Algorithm for Rich Portfolio Optimization. *Appl. Sci.* **2019**, *9*, 3509. [CrossRef]
64. Kalayci, C.B.; Ertenlice, O.; Akyer, H.; Aygoren, H. An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization. *Expert Syst. Appl.* **2017**, *85*, 61–75. [CrossRef]
65. Mendonça, G.H.; Ferreira, F.G.; Cardoso, R.T.; Martins, F.V. Multi-attribute decision making applied to financial portfolio optimization problem. *Expert Syst. Appl.* **2020**, *158*, 113527. [CrossRef]
66. Fernández, E.; Figueira, J.R.; Navarro, J. An interval extension of the outranking approach and its application to multiple-criteria ordinal classification. *Omega* **2019**, *84*, 189–198. [CrossRef]
67. Sunaga, T. Theory of an Interval Algebra and Its Applications to Numerical Analysis. *RAAG Memoirs* **1958**, *2*, 29–46. [CrossRef]
68. Moore, R.E. *Interval Arithmetic And Automatic Error Analysis in Digital Computing*; Stanford University: Stanford, CA, USA, 1963.
69. Hui, E.C.; Chan, K.K.K. Alternative trading strategies to beat "buy-and-hold". *Phys. A Stat. Mech. Its Appl.* **2019**, *534*, 120800. [CrossRef]
70. Hui, E.C.; Chan, K.K.K. A new time-dependent trading strategy for securitized real estate and equity indices. *Int. J. Strateg. Prop. Manag.* **2018**, *22*, 64–79. [CrossRef]
71. Allen, D.E.; Powell, R.J.; Singh, A.K. Chapter 32—Machine Learning and Short Positions in Stock Trading Strategies. In *Handbook of Short Selling*; Gregoriou, G.N., Ed.; Academic Press: San Diego, CA, USA, 2012; pp. 467–478. [CrossRef]
72. Baumann, M.H.; Grüne, L. Simultaneously long-short trading in discrete and continuous time. *Syst. Control. Lett.* **2017**, *99*, 85–89. [CrossRef]
73. Primbs, J.A.; Barmish, B.R. On Robustness of Simultaneous Long-Short Stock Trading Control with Time-Varying Price Dynamics. *IFAC-PapersOnLine* **2017**, *50*, 12267–12272. [CrossRef]
74. O'Brien, J.D.; Burke, M.E.; Burke, K. A Generalized Framework for Simultaneous Long-Short Feedback Trading. *IEEE Trans. Autom. Control.* **2021**, *66*, 2652–2663. [CrossRef]
75. Deshpande, A.; Gubner, J.A.; Barmish, B.R. On Simultaneous Long-Short Stock Trading Controllers with Cross-Coupling. *IFAC PapersOnLine* **2020**, *53*, 16989–16995. [CrossRef]
76. Fu, X.; Du, J.; Guo, Y.; Liu, M.; Dong, T.; Duan, X. A machine learning framework for stock selection. *arXiv* **2018**, arXiv:1806.01743.
77. Zhang, R.; Lin, Z.; Chen, S.; Lin, Z.; Liang, X. Multi-factor Stock Selection Model Based on Kernel Support Vector Machine. *J. Math. Res* **2018**, *10*, 9. [CrossRef]
78. Becker, Y.L.; Fei, P.; Lester, A.M. Stock selection: An innovative application of genetic programming methodology. In *Genetic Programming Theory and Practice IV*; Springer: Berlin, Germany, 2007; pp. 315–334.
79. Levin, A. Stock selection via nonlinear multi-factor models. *Adv. Neural Inf. Process. Syst.* **1995**, *8*, 966–972.

80. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
81. Sengupta, A.; Pal, T.K. On comparing interval numbers. *Eur. J. Oper. Res.* **2000**, *127*, 28–43. [CrossRef]
82. Ishibuchi, H.; Tanaka, H. Multiobjective programming in optimization of the interval objective function. *Eur. J. Oper. Res.* **1990**, *48*, 219–225. [CrossRef]
83. Shi, J.R.; Liu, S.Y.; Xiong, W.T. A new solution for interval number linear programming. *Syst. Eng.-Theory Pract.* **2005**, *2*, 16.
84. Solares, E.; Fernandez, E.; Navarro, J. A generalization of the outranking approach by incorporating uncertainty as interval numbers. *Investig. Oper.* **2019**, *39*, 501–514.
85. Li, G.-D.; Yamaguchi, D.; Nagai, M. A grey-based decision-making approach to the supplier selection problem. *Math. Comput. Model.* **2007**, *46*, 573–581. [CrossRef]
86. Bhattacharyya, R. A grey theory based multiple attribute approach for r&d project portfolio selection. *Fuzzy Inf. Eng.* **2015**, *7*, 211–225.
87. Fernández, E.; Navarro, J.; Solares, E. A hierarchical interval outranking approach with interacting criteria. *Eur. J. Oper. Res.* **2022**, *298*, 293–307. [CrossRef]
88. Ivkovic, N.; Jakobovic, D.; Golub, M. Measuring Performance of Optimization Algorithms in Evolutionary Computation. *Int. J. Mach. Learn. Comput.* **2016**, *6*, 167–171. [CrossRef]
89. McKenna, B. *Why NVIDIA Stock Plunged 31% in 2018*; Motley Fool: Alexandria, VA, USA, 2019.
90. Li, X.; Xie, H.; Chen, L.; Wang, J.; Deng, X. News impact on stock price return via sentiment analysis. *Knowl.-Based Syst.* **2014**, *69*, 14–23. [CrossRef]

symmetry

MDPI

Article

Identifying the Attack Sources of Botnets for a Renewable Energy Management System by Using a Revised Locust Swarm Optimisation Scheme †

Hsiao-Chung Lin [1], Ping Wang [1,*], Wen-Hui Lin [1], Kuo-Ming Chao [2] and Zong-Yu Yang [1]

[1] Faculty of Department of Information Management, Kun Shan University, Tainan 710303, Taiwan; fordlin@mail.ksu.edu.tw (H.-C.L.); linwh@mail.ksu.edu.tw (W.-H.L.); s104001757@g.ksu.edu.tw (Z.-Y.Y.)
[2] School of MIS, Coventry University, Coventry CV1 5FB, UK; k.chao@coventry.ac.uk
* Correspondence: pingwang@mail.ksu.edu.tw; Tel.: +886-6-205-0545

† This paper is an extended version of our paper published in 3rd IEEE Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability 2021 (IEEE ECBIOS2021), Tainan, Taiwan, 28–31 May 2021.

Abstract: Distributed denial of service (DDoS) attacks often use botnets to generate a high volume of packets and adopt controlled zombies for flooding a victim's network over the Internet. Analysing the multiple sources of DDoS attacks typically involves reconstructing attack paths between the victim and attackers by using Internet protocol traceback (IPTBK) schemes. In general, traditional route-searching algorithms, such as particle swarm optimisation (PSO), have a high convergence speed for IPTBK, but easily fall into the local optima. This paper proposes an IPTBK analysis scheme for multimodal optimisation problems by applying a revised locust swarm optimisation (LSO) algorithm to the reconstructed attack path in order to identify the most probable attack paths. For evaluating the effectiveness of the DDoS control centres, networks with a topology size of 32 and 64 nodes were simulated using the ns-3 tool. The average accuracy of the LS-PSO algorithm reached 97.06 for the effects of dynamic traffic in two experimental networks (number of nodes = 32 and 64). Compared with traditional PSO algorithms, the revised LSO algorithm exhibited a superior searching performance in multimodal optimisation problems and increased the accuracy in traceability analysis for IPTBK problems.

Keywords: locust swarm optimisation; distributed denial of service; Internet protocol traceback; multisubswarm strategy; ns-3

Citation: Lin, H.-C.; Wang, P.; Lin, W.-H.; Chao, K.-M.; Yang, Z.-Y. Identifying the Attack Sources of Botnets for a Renewable Energy Management System by Using a Revised Locust Swarm Optimisation Scheme. *Symmetry* **2021**, *13*, 1295. https://doi.org/10.3390/sym13071295

Academic Editor: Jan Awrejcewicz

Received: 26 June 2021
Accepted: 17 July 2021
Published: 19 July 2021

1. Introduction

A series of major information security incidents have occurred recently. Information security hazards include not only individual hackers highlighting their technical capabilities, but also team attacks aimed at obtaining economic benefits. For example, in 2016, the servers of the First Bank of Taiwan were attacked with a trojan horse from the United Kingdom [1]. Several security breaches involving distributed denial of service (DDoS) attacks have occurred in Taiwan. The Financial Services Information Sharing and Analysis Centre, which is the only global cyber intelligence sharing community solely focused on financial services, reported that more than 100 financial services firms were the targets of a wave of DDoS extortion attacks conducted by the same actor in February 2021. These DDoS attacks by botnets resulted in people being unable to place brokerage orders online with the aforementioned firms. The hackers behind the aforementioned attacks demanded a large ransom from the firms, and threatened to detonate the money by using implanted trojans and launch a new wave of DDoS attacks [2].

The Taiwan Stock Exchange announced that, after suffering DDoS attacks, several companies adopted DDoS attack flow cleaning services based on network intrusion prevention systems in 2020. These services provide possible connections to trace the sources of real attacks, analyse the behavioural feature of cyber attacks with data collection [3], and enable countermeasures to be taken against DDoS threats. To counter DDoS attacks, security managers use the Internet protocol (IP) traceback (IPTBK) scheme for periodically detecting and identifying possible threats.

In the identification of the sources of DDoS attacks from botnets, defenders are assumed to have the ability to collect only a small amount of routing information. Therefore, in real-time IP traceability analysis of the botnet command and control (C&C), a small number of router records are required to trace the attack source successfully in the shortest time. In practice, defenders use machine learning algorithms, such as particle swarm optimisation (PSO) [4–7], the genetic algorithm, and ant colony optimisation, to trace the attack source. The routing information of the attack path is used for recursively estimating multiple possible attack paths on the Internet, finding the real attack URL, and marking the compromised host. However, because the traditional PSO algorithm has a nonoptimal balance between path exploration and exploitation in the search strategy, it often provides a suboptimal solution of the target, and often only particles travel on the same attack paths towards the attack sources. Generally, multi-swarm systems provide a new approach to improve this balance based on multi-swarm optimisation. Multi-swarm optimisation uses multiple sub-swarms instead of one swarm, and ensures that each sub-swarm explores a specific region with symmetrical competitive interactions in biology.

Inspired by multi-swarm PSO (MS-PSO) schemes [8–15], the present study used locust search PSO (LS-PSO) to identify the multiple attack sources generated by DDoS attacks from botnets. In this study, DDoS attack paths with a high success probability were reconstructed by marking router packets, tracing the IPs of the botnet C&C by using the LS-PSO algorithm based on multi-swarm optimisation, and preventing spoofed IP attacks.

In summary, the primary contributions of this study are as follows:

- Different botnet attack sources were analysed with the locust search mechanism to solve the multimodal optimisation problem according to the natural behaviour of locust swarms.
- The optimal route-searching process of the LS-PSO algorithm was improved using the short-range force (SRF) of a wave of swarm particles (WOSP) [9,10] to prevent the LS-PSO algorithm from converging prematurely to a local suboptimal solution.
- The LS-PSO algorithm has the advantages of the WOSP algorithm and gradient descent method. The LS-PSO algorithm constructed the attack paths first using the cluster strategy, then the route (CFRS) strategy for path searching in the entire solution space, and then found the global multi-objective solution.
- The accuracy of the LS-PSO algorithm was 99.07% and 95.05% for the effects of dynamic traffic in two experimental networks (number of nodes = 32 and 64, resectively).
- Compared with other route-searching algorithms, such as the A* algorithm [16] and PSO algorithm [4–7], the proposed LS-PSO algorithm exhibited a higher traceback accuracy in the reconstruction of attack paths, which were employed for analysing attack origins from multiple data sources by using ns-3 with the Boston University Representative Internet Topology Generator (BRITE) framework.

The remainder of this paper is organised as follows. Section 2 presents a review of the locust swarm optimisation (LSO) algorithm for solving multimodal optimisation problems. Section 3 describes the LS-PSO scheme for solving the IPTBK problem. Section 4 presents the experimental results obtained using the LS-PSO algorithm with the ns-3 network simulator, and describes the global heuristic performance of the algorithm. Finally, Section 5 concludes the study.

2. Overview of Multiswarm PSO Schemes

This section reviews several existing multi-swarm PSO schemes for identifying the possible sources of DDoS attacks.

The LSO algorithm [14] was proposed by Stephen Chen in 2009. The original concept of the LSO algorithm is based on the optimisation of group actions according to the biological intelligence of birds. In the process of a locust swarm searching for food (best solution), each locust (individual) represents a solution. The process of PSO involves dispersing each locust in a certain solution space, searching each specific space with a locust, and sharing the information found with the entire swarm. The locust swarm updates its movement according to the route information and the previous experiences of the locusts. Such updates in the movement direction enable the entire swarm to search for food successfully, that is, to find the best solution in the entire solution space.

Three multimodal optimisation methods use revised PSO schemes: the WOSP [9,10], dynamic multiswarm particle swarm optimisation (DMS-PSO) [11–13], and LSO algorithms [14,15]. The LSO algorithm is introduced in the following text.

LSO Algorithm

The LSO algorithm is basically a revised version of the WOSP algorithm, which is derived from the PSO algorithm, and aims to prevent premature convergence to local suboptimal solutions associated with the SRF, which comprises the repulsion force (RF) and attraction force (AF; Figure 1) based on symmetric competitive interactions in the biology. In the PSO algorithm, every individual has unique motivations; however, the swarm tends to follow the leader. When newly joined individuals find a better solution on the search path, the leader updates the trajectory. The trajectory of each individual is an inertial route towards the best position known in the passage [9].

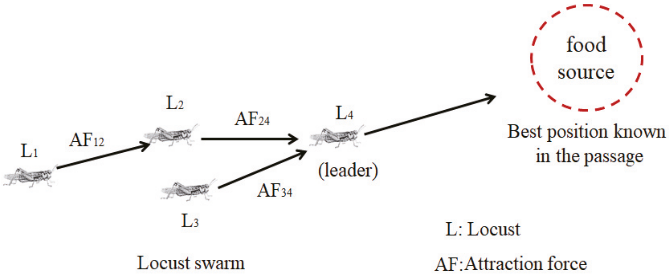


Figure 1. Movement of locust swarms in the WOSP algorithm.

As displayed in Figure 1, the difference between the LSO and PSO algorithms is that the LSO algorithm is designed to solve multimodal optimisation problems, where locusts (subswarms) simultaneously search for multiple objects. The search position is continually updated through iterations by optimising locust swarms to find the optimal solution.

To enhance the search for multiple optima, the LSO algorithm uses the following two main adjustments for finding a new optimal solution when particles converge near the local optima: (1) the SRF is used to disperse the neighbouring particles escaping from the current local optimum and regroup partial particles into new subswarms, and each subswarm explores the possible best solution (Figure 2), and (2) the starting point is optimised by using the evolutionary algorithm in the best-found optimum process. In general, the evolutionary algorithm produces an excellent next generation of particles to adapt to the changing environment through innate inheritance and acquired knowledge. The pseudocode of the LSO algorithm developed by Chen (2009) [14] is presented as follows (Algorithm 1).

Algorithm 1 Locust Swarm Optimisation (LSO) Algorithm

```
1. For Swarm 1
2. Generate $R$ random particles
3. Select a subset $S$ with the best particles for a particle swarm
4. Assign a random velocity to each particle
5. Run each particle for $n$ iterations
6. Optimise the best particles by using the gradient descent search algorithm
7. For Swarms 2–$N$
8. Generate $R$ random points around the previous optimum
9. Select a subset $S$ with the best points for a specific particle swarm
10. Run each particle for $n$ iterations
11. Optimise the best points by using the gradient descent search algorithm
12. Obtain the best optimum
```

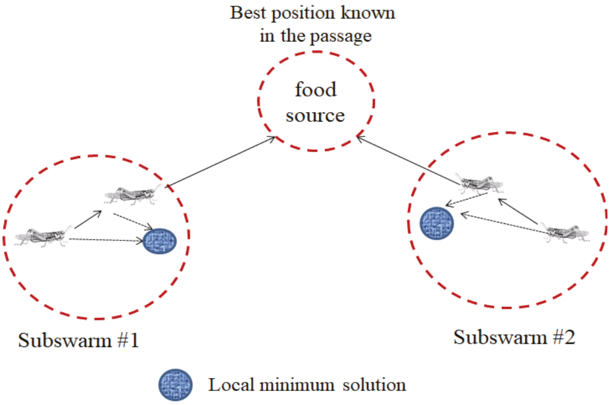


Figure 2. Search process of the proposed LSO algorithm when using two subswarms.

For Swarm 1-, the LSO algorithm imitates the moving effect of biological swarms when the particles are far away from each other. Thus, the mutual AF between the neighbouring particles produces a particle swarm. The AF leads close for the neighbouring particles and accelerates for each other. After a certain period, the distance between the adjacent particles in the swarm is too low. At this time, these particles generate an RF so that they move away from each other before the next time interval.

For Swarms 2–N-, the LSO algorithm uses a "jump to pre-set direction" strategy at a fixed time interval according to the scout particle suggestions in order to allow partial particles to jump from the original subswarms. In the LSO algorithm, the scout particles are used for selecting the starting point of the jump particles to be scattered in order to explore the best solution. However, the rescattering time and particle direction are random in the WOSP algorithm.

The smart starting point is found using the random search strategy (when R > S, R and S are selected randomly from selected particle swarms) [14] to detect new possible solutions. In line 10, the initial velocity of the particles is selected, such that the particle movement is directed away from the previous best solution and previously searched space.

The LSO algorithm has exhibited good results when solving multimodal optimisation problems. Moreover, multiswarm PSO schemes, such as the WOSP, DMS-PSO, and LSO algorithms, have also provided good search results when solving multimodal problems. Table 1 compares these three PSO schemes.

Table 1. Comparisons of the features of three locust search PSO schemes.

Scheme	Feature	Advantage	Limitation
WOSP algorithm [9,10]	To avoid premature convergence, the WOSP algorithm uses the short-range force (SRF) to repel particles that are too close to each other.	The WOSP algorithm is especially suitable for the optimisation of multimodal problems with multiple local optima by using the SRF.	Sometimes, the WOSP algorithm may generate loop iterations in the process of searching for the global optimum.
MS-PSO algorithm [11–13]	The DMS-PSO algorithm periodically regroups the particles of subswarms after they have converged into new subswarms, and new swarms are produced with particles from previous swarms.	By using the local search and convergence search processes, the DMS-PSO algorithm can achieve a good balance between exploration and exploitation abilities in multimodal problems.	The DMS-PSO algorithm separates the optimum search process into two distinct phases, which can weaken both mechanisms of the search process.
LSO algorithm [14,15]	The LSO algorithm uses a fixed time interval and direction according to the scout particles for suggesting the starting point of the multiparticle swarm for scattering to explore possible new paths.	The LSO algorithm selects scout particles to propose smart starting points for multiswarm PSO for scattering. The particle movement is directed away from the previous best solution and previously searched solution space.	The LSO algorithm complicates the search process for the optimal solution.

3. Application of the LSO Algorithm for Solving the IPTBK Problem

The proposed LS-PSO algorithm was used to analyse the accuracy of the attack path reconstruction at various topology sizes. The basic IPTBK problem is described in the following subsections. The problem of the attack path reconstruction can be expressed as a directed graph as follows: $G = (N, E) = (n_i, e_{ij})$, where N represents a set of nodes, $N = [n_i] = \{n_{i1}, \ldots, n_{id}, \ldots, n_{iD}\}$, n_s is a set of nodes for attack sources, n_d represents a set of victims, and E denotes set of edges e_{ij} of the graph from node x_i to node x_j in D-dimensional search space.

3.1. Basic Idea

When solving multimodal optimisation problems, the main aim is to find multiple optimal solutions (global optimum and local optima) associated with a single cost function. Theoretically, multiswarm PSO is suitable for optimisation in multimodal problems with multiple local optima, because it can achieve a good balance between exploration and exploitation behaviours. However, the performance of multiswarm PSO algorithms is dependent on the starting points selected in the search process. When solving multimodal problems, new starting points can be randomly selected or derived from known solutions. In general, starting points are randomly selected. However, many search spaces are globally convex; thus, the quality of the local optima increases as the distance from the global optimum decreases. In the global convex search space of the LSO algorithm, if the starting point is selected near the area of the optimum, the global optimum solution can be obtained using the gradient descent algorithm. To explore possible new solutions, the proposed LS-PSO algorithm uses two behaviours of biological locust swarms, namely, solitary operation and social operation.

Solitary operation. Similar to the behaviour of biological locust swarms, when neighbouring particles are far away from each other within the swarm, the AF ensures group cohesion. Conversely, when neighbouring particles are too close, they are expelled by the SRF into new subswarms, which prevents premature convergence. The SRF can accelerate the particles to separate in different directions at a fixed time interval (Figure 3).

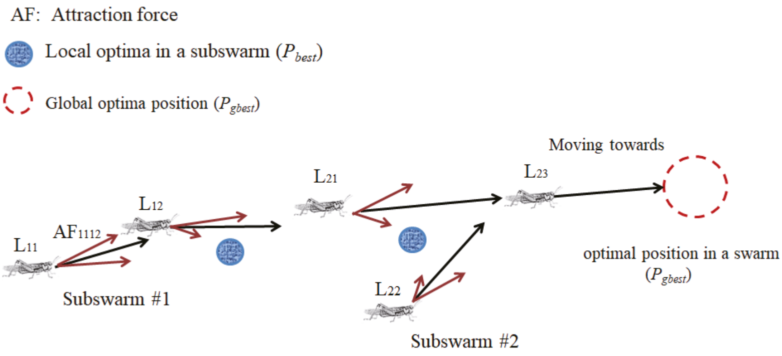

AF: Attraction force
Local optima in a subswarm (P_{best})
Global optima position (P_{gbest})
Moving towards
L23
L21
L12
AF1112
L11
Subswarm #1
optimal position in a swarm
(P_{gbest})
L22
Subswarm #2

Figure 3. Prevent the LS-PSO algorithm from converging prematurely using the SRF.

To improve the searching ability of the LSO algorithm, a scout particle is introduced in a swarm to suggest a search direction. In particular, the scout particle recommends the starting point of the re-searching process to find the best path to food sources at the ending period of the search. Thus, the re-searching process prevents most particle swarms from converging prematurely to a local suboptimal solution when the fitness function value has stabilised.

Social operation. To prevent most particle swarms from converging prematurely to local optima, the search space must be expanded using neighbouring particles to form a new particle swarm according to the social behaviour of locust swarms. In practice, the initial values of the particles of the new subswarms are set as close as possible to the global best solution, which decreases the search time for the regional optimal solution. Thus, the intelligent selection of a starting point effectively reduces the computational time, but maintains the accuracy of the recursive re-searching process. Therefore, this research focused on finding the best starting points for the scattering operation.

When all the particles converge quickly to a single attack path, the particle subswarms are forced to make dynamic changes in the neighbourhood structure, as illustrated in Figure 4. Thus, adjacent subswarms are reorganised by grouping partial particles into new subswarms for expanding the search space of each subswarm. When each subregion is reorganised and generated at each iteration, some of the particles of the subswarms are periodically randomly recombined, and the new subswarms search the adjacent regions again. *R* denotes the reorganisation period. In the aforementioned method, each subswarm can fully exchange information with the other subswarms. Compared with the traditional (static) neighbourhood structure, the new neighbourhood structure has greater freedom, which increases the diversity of the particle swarm searching.

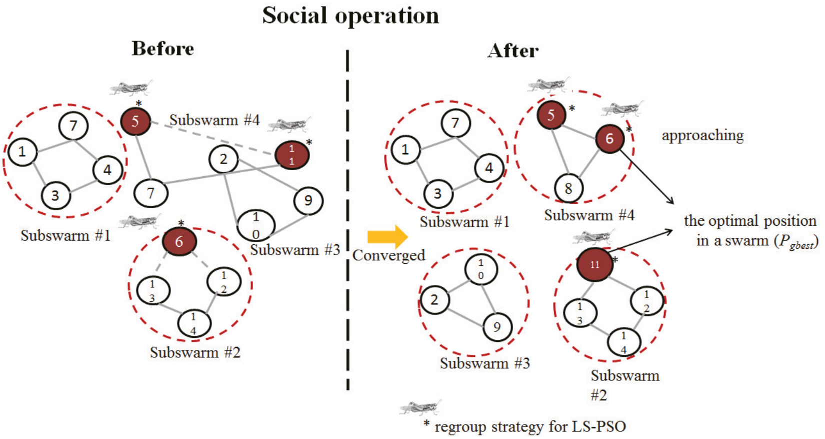


Figure 4. Regroup strategy for multiswarm optimisation in the LS-PSO.

To prevent premature convergence to local optima, two modified approaches are proposed with updated rules for multi-objective searching: (1) multiswarm optimisation and (2) intelligent starting point selection. In multiswarm optimisation, which is inspired by the DMS-PSO method [11–13], the locust swarm periodically regroups the particles of the subswarms after they have converged into new subswarms. The new swarms are produced using particles from previous swarms using the regroup strategy (Figure 4).

In intelligent starting point selection, the starting point is selected near the best area in the global convex search space by using a nonrandom adaptive subswarm scattering strategy. The LSO algorithm attempts to jump using a fixed time interval and direction according to the suggested scout particle scattering at the starting point of the local optimum.

3.2. Tracing the Sources of DDoS Attacks by Using the LS-PSO Algorithm

To prevent particle swarms from converging quickly on a single path, the proposed LS-PSO algorithm divides them into several subswarms. Furthermore, to solve the multimodal search optimisation problem, the rules of each subswarm must be updated in the proposed LS-PSO algorithm.

Assuming that the particle swarm represents a group of packets in the attack path, each packet header record includes the source IP address, the address of the next route, and the destination address. Moreover, we consider that the highest fitness value would be obtained for the most recent experience in which particles travel on the best path. Multiple possible attack paths exist between the nodes (n_i, ..., n_k, ..., n_j). The fitness value of each path is calculated to check whether a particle has travelled on a low-cost path. Usually, the path search algorithm is used to improve the efficiency of a travel routing system by considering the selection of low-cost network routes, that is, where the distance between two nodes (i.e., n_i and n_j) is shorter, the hop count (d_{ij}) is smaller, and the path between the two nodes (n_i, n_j) has a high quality of service (QoS). Therefore, the path search algorithm usually selects the path with the lowest routing cost (i.e., the shortest travel distance and highest QoS to reduce the routing time). In general, the route cost of path C_i from node x_i to the victim is inversely proportional to the distance travelled and directly proportional to the QoS (i.e., a high QoS corresponds to a low transmission delay and low traffic congestion); thus, $Lp = f\left(QoS, \frac{1}{d_{ij}}\right)$. Theoretically, the minimum cost function (*Lp*) must be determined to solve the multipeak optimisation problem; however, this function is subject to routing cost constraints. The minimum cost function (*Lp*) is a positive number ($\sum_{j=1}^{n} C_i.p_{ij}^n > 0$), which is expressed as follows:

$$Fitness = Lp = \sum_{j=1}^{n} C_i.p_{ij}^n, \tag{1}$$

$$Min\ \mathrm{L}p,\ \forall i, j$$

$$Subject\ to\ \sum_{j=1}^{n} C_i.p_{ij}^n > 0, \tag{2}$$

where *Fitness* represents the fitness value of a path. Adaptability is considered to evaluate the suitability of each path, and p_{ij}^n indicates whether a path exists from node *i* to node *j* for particle *n*. An p_{ij}^n value of 1 indicates that a path exists from node *i* to node *j* for particle *x*, and an p_{ij}^n value of 0 indicates that the aforementioned path does not exist.

Route searching approach: In the proposed LS-PSO algorithm, a two-stage route searching approach based on the cluster first, route second (CFRS) strategy is used for path searching in the entire solution space. Our solution technique involves creating subswarms of particles that contain certain information regarding the destination. Inspired by the CFRS strategy used in capacity-constrained vehicle routing problems, this study divided the attack source into multiple network areas according to the IP domain associated with the timing data from DNS logs to determine the minimum cost path to the destination on the basis of a weighted graph theory.

The CFRS performs a single swarming of the vertex set and then determines a route with the minimum cost for each swarm. It also regularly expands possible paths from the destination node by examining the possible paths of the starting node until the end condition is satisfied for reconstructing the overall attack paths. In addition, the CFRS assigns several subswarms of particles in sequence to each local area. It uses heuristic algorithms to acquire the global optimum. The advantage of using FBCFRS is that by clustering the routing traffic, the attack sources can be found within multiple local areas in advance. Moreover, the redundancy of the attack path reconstruction can be reduced.

Exploration and exploitation processes: The exploration and exploitation processes follow different strategies. When solving multimodal optimisation problems, exploration involves following a new route, whereas exploitation involves following an existing route. In the exploitation process, the focus is on determining the local optimal solution by using the local and global updates of the position and velocity vectors. Therefore, the fitness value of each path in each subswarm must be updated to evaluate whether the particles travel on attack paths towards the attack sources. In the exploration stage, the global optimum is found using the regrouping method.

On the basis of previous studies on the use of the LSO algorithm in IPTBK analysis, the operation and verification process of the proposed LS-PSO algorithm is divided into three subphases, as depicted in Figure 5.

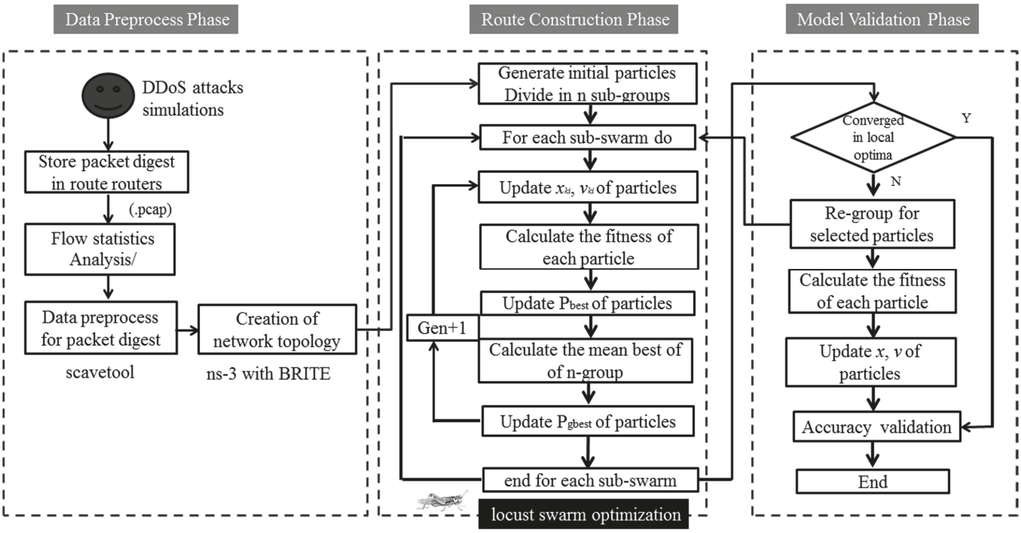

Data Preprocess Phase
Route Construction Phase
Model Validation Phase
DDoS attacks simulations
Store packet digest in route routers
(.pcap)
Flow statistics Analysis/
Data preprocess for packet digest
scavetool
Creation of network topology
ns-3 with BRITE
Generate initial particles Divide in n sub-groups
For each sub-swarm do
Update x_{ki}, v_{ki} of particles
Calculate the fitness of each particle
Update P_{best} of particles
Gen+1
Calculate the mean best of of n-group
Update P_{gbest} of particles
end for each sub-swarm
locust swarm optimization
Converged in local optima
Y
N
Re-group for selected particles
Calculate the fitness of each particle
Update x, v of particles
Accuracy validation
End

Figure 5. Flowchart of the LSO algorithm for the analysis of the network DDoS source tracing.

(1) Data preprocessing phase: The tcpdump tool is used to filter and collect the network routing packets required for path exploration, and to mark these packets for subsequent analysis and reorganisation. Then, Unicast Reverse Path Forwarding is used to check each router that passes through it. The source IP of the packet header is used to determine the path of the transmission connection.

(2) Route reconstruction phase: In multimodal optimisation methods, the exploration and exploitation processes are generally performed in different stages [13]. In the solitary operation stage, each subswarm is used to explore possible solutions. In this stage, the proposed LS-PSO algorithm focuses on determining the local optimal solutions in the solution space and prevents particle swarms from rapidly converging on a single path. In the social operation stage, the global optimal position is determined through a regrouping strategy.

2.1 Solitary operation: To increase the search efficiency of the particle swarms, the proposed LS-PSO algorithm divides them into several subswarms when solving multimodal optimisation problems, where the local update rules of each subswarm must be determined. In the LS-PSO algorithm, attack paths are explored and reconstructed on the basis of the route packets collected from the victim to calculate the fitness of each path.

The first particle swarm generates R = 5000 particles, and each subswarm has 20 particles (S = 20). The initial speed set for the LS-PSO algorithm is the same as that set for the LSO algorithm [14].

$$v_o = c_1 \cdot \left(\frac{Range}{2}\right) \cdot (c_2 \cdot rand() - 1) \tag{3}$$

where c_1 and c_2 represent acceleration constants ($c_1 = 0.5$ and $c_2 = 2$), *Range* represents a unit value of particle position updating between the particle position and the centre of the subswarm, and *rand()* represents a random number in the range (0, 1). Suitable acceleration constants can control the particle speed. Route construction is performed using a velocity state updating rule for conducting position updates over 500 iterations (n = 500). The particle position for each iteration is updated using Equations (4) and (5).

$$x_i^k(t) = x_i^k(t-1) + G \cdot v_i^k\ (t-1) \tag{4}$$

To examine whether any particle exists in a particle swarm, the LSO algorithm explores the best position of the particle swarm (P_{best}) by using the gravity vector G (G = (0.95, 0.05)) [14] in Equation (4). Thus, the gravity force attracts all single particles to search the solution space. In the LSO algorithm, a fixed speed ratio of 0.95 is used to update the distance without considering the effects of the network capability (i.e., the node distance (d_{ij}) and QoS). Consequently, determining the best route between two edge nodes is difficult, and most particles travel on the frequently travelled paths. Therefore, the current study considered two important factors, namely d_{ij} and the QoS (Equation (5)).

$$x_i^k(t) = x_i^k(t-1) + \Delta\tau_{ij}^k(t-1) \tag{5}$$

$$\Delta\tau_{ij}^k(t-1) = \begin{cases} \frac{v_i^k(t).QoS}{d_{ij}^k} & for\ the\ optimal\ path\ of\ subswarm\ k \\ 0 & otherwise \end{cases}$$

In Equation (5), $\Delta_{ij}^k(t-1)$ represents the movement of Δt, which is inversely proportional to the path distance d_{ij}^k between the two end nodes. The parameter d_{ij}^k represents the number of hops on the ith attack path in the kth subswarm.

For each iteration i, the new position of each particle is updated using Equation (6).

$$v_i^k(t) = w_i.\ v_i^k\ (t-1) + (1-w_{\rm i})\cdot\left(p_{best} - x_i^k({\rm t}-1)\right) \tag{6}$$

A high w_i value enables the particles to cross the destination easily; however, a small w_i value leads to slow convergence. The LS-PSO algorithm uses the gradient descent algorithm to search for the optimal acceleration factor.

To determine the acceleration factor w_i (Equation (6)), this study used a greedy local search technique associated with the quasi-Newtonian gradient descent method (BFGS) to identify possible local optima with an intelligent reconnaissance strategy (Equation (7)). Theoretically, the BFGS algorithm can efficiently search for the optimal particle positions when the particle is alone (P_{best}) and in a subswarm (P_{gbest}) in a convex space. Moreover, it can efficiently improve the solution quality of each particle. To determine P_{best} and P_{gbest} for a subswarm, the BFGS algorithm can be used for dynamically adjusting the particle acceleration (weight: w_i) to avoid overfitting by minimising the routing cost C_i (Equation (7)).

$$w_i(t+1) = w_i(t+1) - \eta\frac{\partial C_i}{\partial w_i} \tag{7}$$

where η is the learning factor.

The recursive process with the aforementioned updating rule generates P_{best} and P_{gbest} values for estimating the fitness value for each particle. The fitness value of each particle is calculated to examine whether the particle selects the best route. When a particle moves to a new position, the fitness value is calculated for this position. If the fitness value for the new position is higher than that for the previous best position (i.e., P_{best}), the value of P_{best} must be replaced by the fitness value for the new position, updated according to the particle's optimal experience. Similarly, P_{gbest} must be replaced by P_{best} if the fitness value of the new position is higher than P_{gbest}.

2.2 Social operation: To prevent the majority of subswarms from converging quickly to local optima, the LS-PSO algorithm uses the regrouping strategy to enable particles to escape from the original subswarms because of the mutual RF between particles. A fraction (e.g., 30%) of the particle subswarm is randomly selected to form a new subswarm. In the new subswarms, the starting points of the jumping particles are maintained around the best position P_{gbest} so as to improve the search results in the social operation process. The particle position is updated as follows:

$$x_i^k(t) = x_i^k(t-1) + \Delta x_i^k(t-1)\Delta x_i^k(t-1) = \pm Range * (1 + |rand() * spacing|) \tag{8}$$

where *Range* represents a unit value of particle position updating between the particle position and the centre of the subswarm. The random jump distance is set using the term $|rand\ () * spacing|$, for example, set spacing = 0.3 for small variation. The initial velocity is set using Equation (9) to accelerate the particles away from the previous local optima.

$$v_0^k(t) = v_o + v_i^k\ (t-1)\cdot\left(x_i^k(\mathrm{t}-1) - P_{best}\right) \qquad (9)$$

where v_0^k is shown in Equation (3).

The original subswarm and new subswarm then restart the search process and continue searching until the cost function error is less than the pre-set value or the maximum number of iterations is reached.

(3) Model validation phase: After updating the velocities and positions of the particle swarms, the proposed LS-PSO algorithm must determine the best path for successfully tracing the sources of DDoS attacks. The model accuracy is evaluated using the coverage percentage (%), which is the ratio of the average number of packets on an attack path to the total number of routing packets. The coverage percentage is expressed as follows:

$$\text{Coverage percentage (\%)} = \text{Average number of packets on an attack path/Total number of routing packets}, \qquad (10)$$

where the average number of packets on an attack path is computed as the total number of packets on the route divided by the routing distance (in terms of the hop count). If the converged solution is not the true attack node, then the average number of packets on the route is reset to 0 and the search for the true route is resumed. The complete process is summarised as follows (Algorithm 2).

Algorithm 2 Pseudocode of the LS-PSO Algorithm

```
Input: parameters of the LS-PSO model, including the initial values of max_gen, $c_1$, $c_2$, $x$, and $v$
for the particles, and the network topology generated using Waxman theory
Generate $R$ random points ($x$, $y$) in a swarm
Select a subset $S$ with the best points from the original swarm
Assign an initial velocity to each point by using Equation (3)
While (the number of max_gen) do
For each particle $i$ in the subswarm $k$, do
For each particle $i$, do
Update the velocity $v^{\mathrm{k}}{}_i$ and position $x^{\mathrm{k}}{}_i$ by using Equations (4)–(6)
    Adjust the particle acceleration (weight: $w_i$) by minimising the routing cost $C_i$ by using
Equation (7)
    End for
Calculate the particle fitness value of $x^{\mathrm{k}}{}_i$
End for
If (generation $\geq R$) the neighbouring subswarms are randomly regrouped
    Generate $R$ random points ($x$, $y$) around local optimal solutions
    Select a subset $S$ with the best points from the original swarm
    Set the velocity for each point so that each individual point moves away from its original
location
    Optimise the best point by using Equations (8) and (9)
    Update the individual best position $P_{best}$ and swarm best position $P_{gbest}$
    gen = gen + 1
    End
    Calculate the coverage percentage of each path by using Equation (10)
Output: optimal solution for possible paths
```

4. Discussion

In this study, the ns-3 simulation tool was used to identify botnet C&Cs for countering DDoS attacks and to examine the success probability of DDoS attacks along different attack paths for tracing attack sources in a distributed network. The following situation was simulated: an attacker uses a fake IP address to conduct DDoS attacks and analyses the success probability of the attack source. The applicability of the proposed LS-PSO–IPTBK model was examined using two botnet examples.

For security concerns in academic networks, simulations were performed using the ns-3 software with the BRITE framework on a personal computer with a 2.7-GHz Intel Dual-Core computer processing unit equipped with 4 GB of DDR3 RAM running on Debian 10.9.0 Stable. The simulation of the network security is a cost-effective method for evaluating, testing, and selecting a suitable algorithm. In the simulations, defenders could examine all of the routing options for DDoS attacks and evaluate the basic performance of IPTBK algorithms.

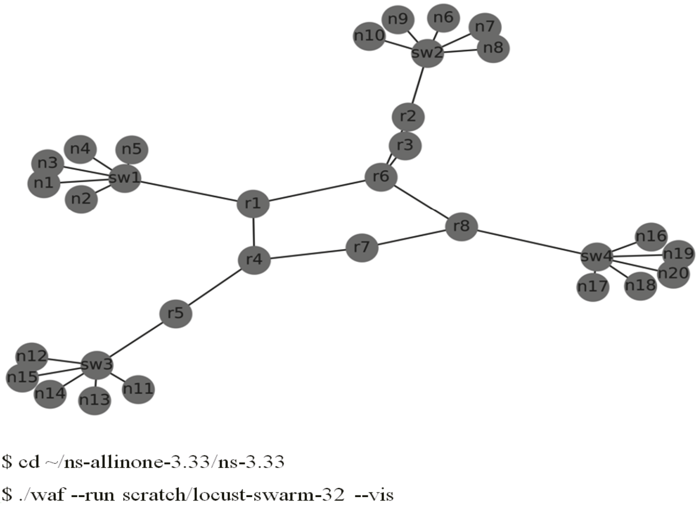
4.1. Case Study I: Network Performance Analysis for DDoS Attacks (32 Nodes)

The first example considers the profiles of DDoS attacks on Internet of Things (IoT) devices on a cloud server. A network intrusion detection system was constructed using the following three processes: (1) data pre-processing, (2) attack path reconstruction, and (3) model validation. The workflow of security analysis is illustrated in Figure 4.

Step 1: Data preprocessing

4.1.1. Creation of the Network Topology

The ns-3 software was deployed with the BRITE framework to generate 32 nodes with integer position coordinates over a rectangular area of 300 × 300, as displayed in Figure 6. As depicted in Figure 6, the simulated network topology consisted of two local area networks (LANs). Simulated hosts and routers were configured using a BriteTopologyHelper class. Furthermore, each of the four LANs had six host nodes, one switch node, one router node, and the relay nodes of the Internet. The attack sites were compromised IoT devices (host 1–host 5 in LAN 1). The switch node was designated as Switch 1. The victim was an online game server (host 11) in LAN 2 (host 6–host 10). We used the Python package networkx to construct the network topology. Each pair of adjacent nodes was an edge that was assigned a weight or cost in all paths. The function attribute res_cost(x,y) was used to indicate the bandwidth and QoS of a path.



```
$ cd ~/ns-allinone-3.33/ns-3.33
$ ./waf --run scratch/locust-swarm-32 --vis
$ ./waf --run "scratch/locust-swarm-32 --tracing=true" --vis (topology.tr)
```

Figure 6. Simulated network topology specified by the BRITE framework (number of nodes = 32).

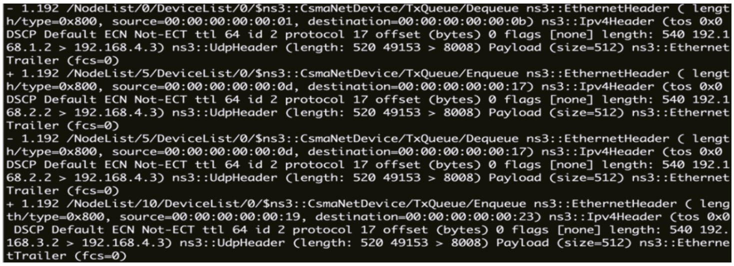
The routing cost of each route in the network topology must be set to decide the next hop path by using the command res_cost = array ([x, y], weight = x). The lower the routing cost of a path, the higher the priority of a packet on it. A high-priority packet can traverse a low-cost path with a relatively small delay.

Step 1.2: Data pre-processing for DDoS threats

Attack Paths Were Constructed Using the Following Two-Step Procedure

Step 1.2.1: Attack on the victim

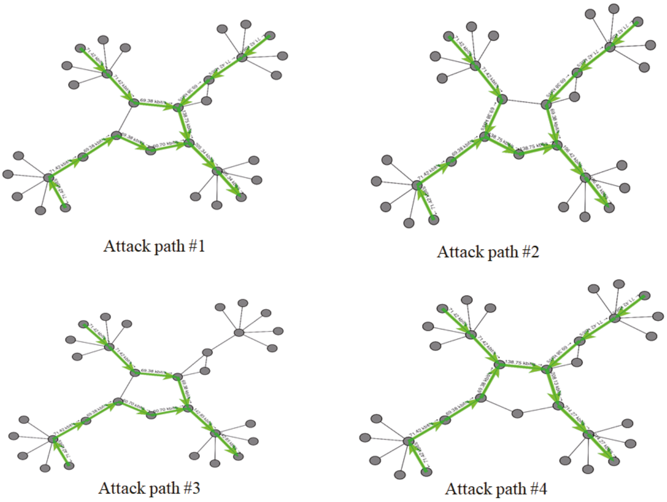
In this step, the attack nodes 1, 6, and 11 (IP addresses of 192.168.1.1, 192.168.1.2, and 192.168.1.3, respectively) launched a series of low-rate DDoS attacks by using UDP floods against the online game server (host 17) in LAN 4. The victim (IP address of 192.168.4.3) listened by default on port 8008. Three cycles of attacks were conducted in 60 s to generate routing information on the victim node for conducting IPTBK, as illustrated in Figure 7.



```
- 1.192 /NodeList/0/DeviceList/0/$ns3::CsmaNetDevice/TxQueue/Dequeue ns3::EthernetHeader ( lengt
h/type=0x800, source=00:00:00:00:00:01, destination=00:00:00:00:00:0b) ns3::Ipv4Header (tos 0x0
DSCP Default ECN Not-ECT ttl 64 id 2 protocol 17 offset (bytes) 0 flags [none] length: 540 192.1
68.1.2 > 192.168.4.3) ns3::UdpHeader (length: 520 49153 > 8008) Payload (size=512) ns3::Ethernet
Trailer (fcs=0)
+ 1.192 /NodeList/5/DeviceList/0/$ns3::CsmaNetDevice/TxQueue/Enqueue ns3::EthernetHeader ( lengt
h/type=0x800, source=00:00:00:00:00:0d, destination=00:00:00:00:00:17) ns3::Ipv4Header (tos 0x0
DSCP Default ECN Not-ECT ttl 64 id 2 protocol 17 offset (bytes) 0 flags [none] length: 540 192.1
68.2.2 > 192.168.4.3) ns3::UdpHeader (length: 520 49153 > 8008) Payload (size=512) ns3::Ethernet
Trailer (fcs=0)
- 1.192 /NodeList/5/DeviceList/0/$ns3::CsmaNetDevice/TxQueue/Dequeue ns3::EthernetHeader ( lengt
h/type=0x800, source=00:00:00:00:00:0d, destination=00:00:00:00:00:17) ns3::Ipv4Header (tos 0x0
DSCP Default ECN Not-ECT ttl 64 id 2 protocol 17 offset (bytes) 0 flags [none] length: 540 192.1
68.2.2 > 192.168.4.3) ns3::UdpHeader (length: 520 49153 > 8008) Payload (size=512) ns3::Ethernet
Trailer (fcs=0)
+ 1.192 /NodeList/10/DeviceList/0/$ns3::CsmaNetDevice/TxQueue/Enqueue ns3::EthernetHeader ( leng
th/type=0x800, source=00:00:00:00:00:19, destination=00:00:00:00:00:23) ns3::Ipv4Header (tos 0x0
 DSCP Default ECN Not-ECT ttl 64 id 2 protocol 17 offset (bytes) 0 flags [none] length: 540 192.
168.3.2 > 192.168.4.3) ns3::UdpHeader (length: 520 49153 > 8008) Payload (size=512) ns3::Etherne
tTrailer (fcs=0)
```

Figure 7. Packet information in a series of low-rate DDoS attacks launched from nodes 1, 6, and 11.

A total of 2685 attack packets (m = 2685) were sent to host 17 by using UDP floods. The average packet quantity of the visited node was the basis for updating the number of particles and assisting particle swarms to trace the sources of attacks by reconstructing the routes (Figure 8).



Attack path #1

Attack path #2

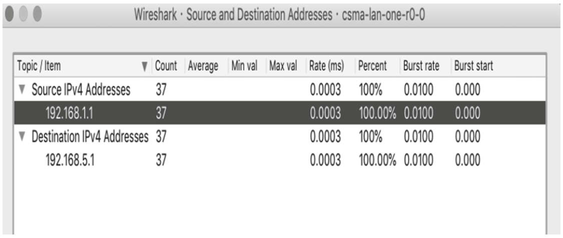
Attack path #3

Attack path #4

Figure 8. Four attack paths for DDoS attacks from nodes 1, 6, and 11 (number of nodes = 32).

Step 1.2.2: Data collection

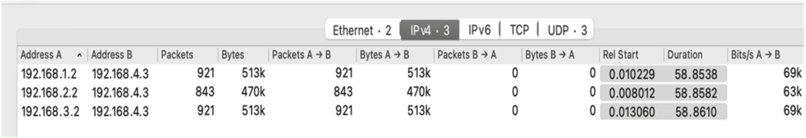
We used Wireshark to collect the samples of network traffic flows from port 8008 of the victim for periodically collecting the routing information of DDoS attacks from routers, as displayed in Figures 9 and 10. The traffic flows were recorded in the Pcap format. After collecting the attack flow packets, scavetool was used to convert the recorded files to the comma-separated values format to the comma-separated values (CSV) format for the reconstruction of attack paths.



Wireshark · Source and Destination Addresses · csma-lan-one-r0-0

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
▼ Source IPv4 Addresses	37				0.0003	100%	0.0100	0.000
192.168.1.1	37				0.0003	100.00%	0.0100	0.000
▼ Destination IPv4 Addresses	37				0.0003	100%	0.0100	0.000
192.168.5.1	37				0.0003	100.00%	0.0100	0.000

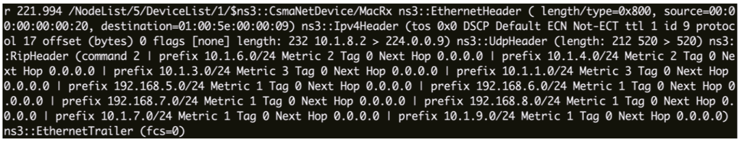
Figure 9. Routing information collected from port 8008 using Wireshark.



Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B
192.168.1.2	192.168.4.3	921	513k	921	513k	0	0	0.010229	58.8538	69k
192.168.2.2	192.168.4.3	843	470k	843	470k	0	0	0.008012	58.8582	63k
192.168.3.2	192.168.4.3	921	513k	921	513k	0	0	0.013060	58.8610	69k

Figure 10. Routing information summarized from the victim.

In practice, the defender uses the *traceroute* command to periodically validate the routing information of DDoS attacks from routers and decide the route from a given source by collecting the sequence of hops the packet traversed, as shown in Figure 11.



```
r 221.994 /NodeList/5/DeviceList/1/$ns3::CsmaNetDevice/MacRx ns3::EthernetHeader ( length/type=0x800, source=00:0
0:00:00:00:20, destination=01:00:5e:00:00:09) ns3::Ipv4Header (tos 0x0 DSCP Default ECN Not-ECT ttl 1 id 9 protoc
ol 17 offset (bytes) 0 flags [none] length: 232 10.1.8.2 > 224.0.0.9) ns3::UdpHeader (length: 212 520 > 520) ns3:
:RipHeader (command 2 | prefix 10.1.6.0/24 Metric 2 Tag 0 Next Hop 0.0.0.0 | prefix 10.1.4.0/24 Metric 2 Tag 0 Ne
xt Hop 0.0.0.0 | prefix 10.1.3.0/24 Metric 3 Tag 0 Next Hop 0.0.0.0 | prefix 10.1.1.0/24 Metric 3 Tag 0 Next Hop
0.0.0.0 | prefix 192.168.5.0/24 Metric 1 Tag 0 Next Hop 0.0.0.0 | prefix 192.168.6.0/24 Metric 1 Tag 0 Next Hop 0
.0.0.0 | prefix 192.168.7.0/24 Metric 1 Tag 0 Next Hop 0.0.0.0 | prefix 192.168.8.0/24 Metric 1 Tag 0 Next Hop 0.
0.0.0 | prefix 10.1.7.0/24 Metric 1 Tag 0 Next Hop 0.0.0.0 | prefix 10.1.9.0/24 Metric 1 Tag 0 Next Hop 0.0.0.0)
ns3::EthernetTrailer (fcs=0)
```

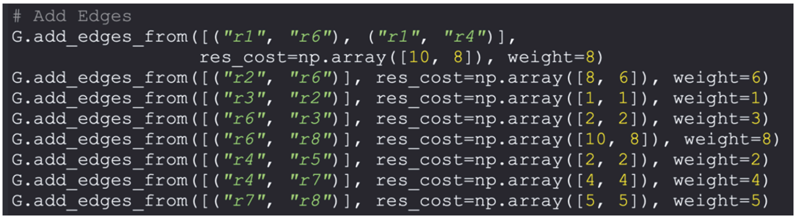
Figure 11. Using traceroute to validate the routing information of DDoS attacks.

Step 2: Route construction

The routing information generated in Step 1.2.2 was used as the input dataset of the PSO model. The main characteristics of the LS-PSO model were as follows: (1) the particle population was set equal to the number of packets collected related to DDoS attacks; (2) the number of generations was set to 500, and the route-searching rules were updated for each generation; (3) the initial value of w_i (weighting factor) was 0.8; (4) c_1 and c_2 were set as 2.0 in Equation (4); and (5) the number of subswarms was 4. For the first particle swarm generated R was set to 5000 particles, and each subswarm had 20 particles (S = 20). Moreover, each particle was run for 500 iterations (n = 500).

4.1.2. Dynamic Routing Costs of All the Routes

Considering the factors of traffic dynamics, including the bandwidth, traffic delay, and QoS requirements in the network transmission, we set different weights (i.e., cost) for each route in the network topology, and decided the next hop path by using the command res_cost = array ([x, y], $w = x$) for paths *r1–r6–r8* and *r1–r4–r7–r8*. Figure 12 indicates that all paths had different routing costs and weights. The higher the routing cost, the lower the routing priority. The larger the weight (w), the larger the bandwidth. Moreover, the shorter the routing distance for a path, the higher the QoS.



```
# Add Edges
G.add_edges_from([("r1", "r6"), ("r1", "r4")],
                  res_cost=np.array([10, 8]), weight=8)
G.add_edges_from([("r2", "r6")], res_cost=np.array([8, 6]), weight=6)
G.add_edges_from([("r3", "r2")], res_cost=np.array([1, 1]), weight=1)
G.add_edges_from([("r6", "r3")], res_cost=np.array([2, 2]), weight=3)
G.add_edges_from([("r6", "r8")], res_cost=np.array([10, 8]), weight=8)
G.add_edges_from([("r4", "r5")], res_cost=np.array([2, 2]), weight=2)
G.add_edges_from([("r4", "r7")], res_cost=np.array([4, 4]), weight=4)
G.add_edges_from([("r7", "r8")], res_cost=np.array([5, 5]), weight=5)
```

Figure 12. Dynamic routing cost settings for paths *r1–r6–r8* and *r1–r4–r7–r8*..

As presented in Table 2, the simulated system comprised eight simple routes between n_1, n_6, and n_{11} (the attack nodes) and n_{17} (the victim). This information was obtained using the all_simple_paths application programming interface call in the networkx suite of ns-3.

Table 2. Set of simple routes between the attack nodes and the victim (n_{11}).

	Simple Routes Using All_Simple_Paths API
Route 1	n_1–Switch 1–router 1–router 6–router 8–Switch 4–n_{17}
Route 2	n_1–Switch 1–router 1–router 7–router 8–Switch4–n_{17}
Route 3	n_6–Switch 2–router 2–router 6–router 1–router 4–router 7–router 8–Switch 4–n_{17}
Route 4	n_6–Switch 2–router 2–router 6–router 8–Switch 4–n_{17}
Route 5	n_6–Switch 2–router 2–router 3–router 6–router 1–router 4–router 7–router 8–Switch 4–n_{17}
Route 6	n_6–Switch 2–router 2–router 3–router 6–router 8–Switch 4–n_{17}
Route 7	n_{11}–Switch 3–router 5–router 4–router 1–router 8–Switch 4–n_{17}
Route 8	n_{11}–Switch 3–router 5–router 4–router 7–router 8–Switch 4–n_{17}

Using the A* search algorithm.

To verify the optimal path of DDoS attacks, the shortest route between the attack nodes and the victim was determined using A* algorithm [16]. The A* algorithm is an improved version of Dijkstra's algorithm. The A* algorithm can be used to identify the shortest path between any two end nodes in a search space. We compared the performance of the A* and LS-PSO algorithms for solving the IPTBK problem. First, we used the A* algorithm in the networkx suite to examine the situation in which n_1, n_6, and n_{11} attacked n_{17}. By using this algorithm, the following paths in Table 3 were identified as having the lowest costs.

Table 3. Set of the shortest routes between the attack nodes and the victim (n_{11}).

	Routes between the Attack Nodes and the Victim Using A* Search Algorithm
Route 1	n_1–Switch 1–router 1–router 6–router 8–Switch 4–n_{17}
Route 2	n_6–Switch 6–router 2–router 6–router 8–Switch4–n_{17}
Route 3	n_{11}–Switch 3–router 5–router 4–router 7–router 8–Switch 4–n_{17}

Using the LS-PSO algorithm.

We used the LS-PSO algorithm to analyse the possible attack path for the attack case $n_1 \rightarrow n_{17}$. By reconstructing the attack path for this case, we applied the subgroup searching strategy to obtain the optimal solution. Particles travelled around all the paths and back to the attack origins according to the local and global updating rules presented in Equations

(5)–(10). After 500 generations had been executed, the results revealed two possible attack paths for conducting model performance analysis.

Route 1: [$'n_{1'}, 'sw1', 'r1', 'r6', 'r8', 'sw4', 'n_{17'}$]
Route 2: [$'n_{1'}, 'sw1', 'r1', 'r4', 'r7', 'r8', 'sw4', 'n_{17'}$]

Similarly, the following paths were obtained for the attack cases $n_6 \rightarrow n_{17}$ and $n_{11} \rightarrow n_{17}$.

Route 3: [$'n_{6'}, 'sw2', 'r2', 'r6', 'r8', 'sw4', 'n_{17'}$]
Route 4: [$'n_{11'}, 'sw3', 'r5', 'r4', 'r7', 'r8', 'sw4', 'n_{17'}$]
Route 5: [$'n_{11'}, 'sw3', 'r5', 'r4', 'r1', 'r6', 'r8', 'sw4', 'n_{17'}$]

Step 3: Model validation phase

After 500 generations had been executed, the coverage rate of the attack path in the experimental case was calculated using Equation (10) (Table 4). The first three paths presented in Table 4 were selected as the possible attack paths for the experimental case, in which the minimum support threshold was t = 3%. As presented in Table 4, the LS-PSO–IPTBK model exhibited an accuracy of 99.07% (m = 2685) for static traffic; thus, the error rate was 0.93% for the network topology (number of nodes = 32).

Table 4. Possible paths of DDoS attacks (number of nodes = 32).

	Attack Path	Packets Collected	Coverage Percentage
Route 1	n_1–Switch 1–router 1–router 6–router 8–Switch 4–n_{17}	840	31.29%
Route 2	n_1–Switch 1–router 1–router 4–router 7–router 8–Switch 4–n_{17}	840	31.29%
Route 3	n_6–Switch 2–router 2–router 6–router 8–Switch 4–n_{17}	810	30.17%
Route 4	n_{11}–Switch 3–router 5–router 4–router 7–router 8–Switch 4–n_{17}	70	2.06%
Route 5	n_{11}–Switch 3–router 5–router 4–router 1–router 6–router 8–Switch 4–n_{17}	100	3.12%
Total		2660	99.07%

4.2. Case study II: Network Performance Analysis for DDoS Attacks (64 Nodes)

In the second experiment, a series of DDoS attacks were conducted in a simulated network topology (number of nodes = 64) using ns-3 with BRITE to evaluate the convergence performance of the proposed model. As shown in Figure 13, the simulated network topology consists of eight local area networks (LANs). In the following, the attack nodes, including n_1, n_{11}, n_{21}, n_{31}, and n_{36}, launched a series of low-rate DDoS attacks using UDP floods against the node17 (port 8008) in LAN4. Six cycles of attacks were conducted in 60 s and a total of 4526 attack packets (m = 4526) were sent to host 17 using UDP floods, as illustrated in Figure 14.

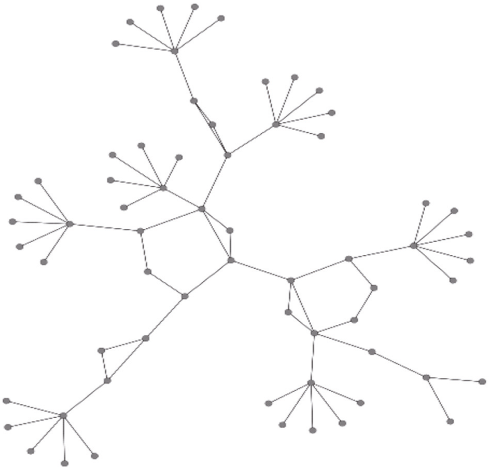


Figure 13. Simulated network topology specified by the BRITE framework (number of nodes = 64).

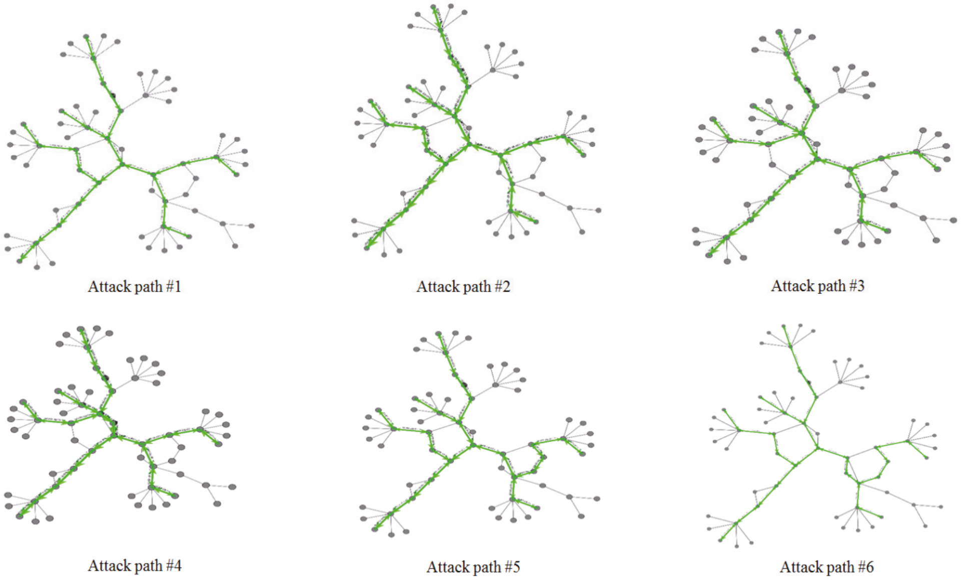


Figure 14. Six attack paths for Distributed Denial-Of-Service (DDoS) attacks from nodes 1, 11, 21, 31, and 36.

Similar to the first experiment, we used the LS-PSO algorithm to analyse the possible attack path for the test case ($n_1 \rightarrow n_{17}$), ($n_{11} \rightarrow n_{17}$), ($n_{21} \rightarrow n_{17}$), ($n_{31} \rightarrow n_{17}$), and ($n_{36} \rightarrow n_{17}$). The coverage rate of the attack path in the experimental case was calculated using Equation (10), and the experiment results for the five cases are listed in Table 5.

Table 5. Possible paths of DDoS attacks (number of nodes = 64).

	Attack Path	Packets Collected	Coverage Percentage
Route 1	*n1-switch1-router1-router6-router9-router8-router10-switch4-n17*	508	11.22%
Route 2	*n1-switch1-router1- router13-router6-router9-router8-router10-switch4-n17*	508	11.22%
Route 3	*n11-switch3-router7-router5-router6-router9-router8-router10-switch4-n17*	420	4.46%
Route 4	*n21-switch5-router17-router4-router2-router1-router6-router9-router8-router10-switch4-n17*	202	9.28%
Route 5	*n21-switch5-router17-router2-router1-router6-router9-router8-router10-switch4-n17*	428	9.46%
Route 6	*n31-switch7-router18-router11-router12-router7-router5-router6-router9-router8-router10-switch4-n17*	426	9.41%
Route 7	*n31-switch7-router18-router11-router12-router7-router14-router5-router6-router9-router8-router10-switch4-n17*	423	9.35%
Route 8	*n26-switch6-router19-router1-router13-router6-router9-router8-router10-switch4-n17*	456	10.08%
Route 9	*n26-switch6-router19-router16-router9-router8-router10-switch4-n17*	503	11.11%
Route 10	*n31-switch7-router18-router5-router6-router9-router8-router10-switch4-n17*	428	9.35%
Total		4258	95.05%

Table 5 indicates the LS-PSO accuracy considering that the static traffic was 95.05% (m = 4526) and that the error rate was 4.95% for the network topology (number of nodes = 32).

The effect of the network size on the number of packets required to construct the attack path was also investigated. Table 6 shows the accuracy and execution time for a test set of routing algorithms with different topology sizes. The experimental results indicate that the execution time of PSO algorithm is higher than that of the LS-PSO algorithm due to the global optimal position that is exploited using the regrouping strategy in the LS-PSO algorithm. Furthermore, the experimental results indicate that the traceback error decreased as the size of the testing data increased.

Table 6. Traceback accuracy vs. execution time of DDoS attacks using the proposed algorithm with A* and PSO algorithm.

Topology / Scheme	***ns* = 32 Nodes**	***ns* = 64 Nodes**
A* search algorithm	90.15%/0.66 ms	87.75%/1.88 ms
PSO	93.16%/66,629.38 ms	90.04%/108,499.41 ms
LS-PSO	99.07%/28,587.81 ms	95.05%/56,657.07 ms

Obviously, the accuracy of the proposed algorithm is higher than those of the A* and the PSO algorithm. Two experimental results demonstrated that the traceback error increased with an increase in the size of the network topology (*ns*) for the LS-PSO algorithm. The overall accuracy rate for the two test cases was 97.06%.

4.3. Case Study III: Network Performance Analysis for DDoS Attacks with Different Subswarms (64 Nodes)

In practice, the LS-PSO algorithm needs to determine the best number of subswarms. The effect of the search strategy with different numbers of subswarm particles (*ns* = 2, 4, and 8) on the number of packets required to construct the attack path in a medium-scale network topology was also investigated. For similar test runs, a series of DDoS attacks was conducted on two simulated network topologies (number of nodes = 32 and 64) to evaluate the convergence performance of the proposed model (Figures 10 and 12). In the experiment, the attacker flooded the victim with packets originating from the attack nodes.

A total of 4526 attack packets were sent in irregular bursts within 120 s to congest the link. Table 7 presents the coverage percentage achieved with the proposed model in the experiment. The results presented in Table 7 indicate that the accuracy of the LS-PSO algorithm was 97.96% when the number of swarms was 2 (*R* = 5000, *S* = 20, and 500 generations); thus, the corresponding error rate was 2.04%. Moreover, the accuracy of the aforementioned algorithm was 98.29% and 97.60% when using four and eight swarms, respectively. The experimental results indicated that the LS-PSO scheme achieved a higher accuracy for medium-scale networks when using four-swarm than when using two-swarm or eight-swarm.

Table 7. Experimental results obtained with different numbers of subswarms.

Topology Scheme	*ns* = 32 Nodes	*ns* = 64 Nodes
A* search algorithm	90.15%	87.75%
PSO	93.16%	90.04%
LS-PSO(2)	99.27%	96.65%
LS-PSO(4)	99.42%	97.15%
LS-PSO(8)	99.08%	96.12%

5. Conclusions

This paper presents an LS-PSO algorithm for solving the IPTBK problem. The proposed algorithm can analyse the effects of multiple-swarm search strategies on the quality of PSO solutions to improve the reconstruction accuracy for the probable paths of DDoS attacks. The experimental results confirmed that the proposed algorithm can analyse the possible attack paths of botnets and the attack sources of DDoS threats by IP traceback techniques with the LS-PSO algorithm.

Although the proposed algorithm can solve the IPTBK problem, its use entails practical challenges. For example, in the case of a DDoS attack, the proposed algorithm generates unusually high rates of packet loss across a network, which may disorder the sequence of packet marking and thus may reduce the traceback accuracy of the algorithm. To implement rapid countermeasures against cyber-attacks involving DDoS flooding in high-speed networks, such as 5G networks, partial packets with high rates of packet loss must be collected. A future study will examine the minimum number of packets required to determine the origins of attacks.

Author Contributions: Conceptualization, P.W.; methodology H.-C.L. and K.-M.C.; resources, P.W.; formal analysis, H.-C.L.; data curation, H.-C.L.; writing—original draft, H.-C.L. and W.-H.L.; writing—review and editing, P.W. and K.-M.C.; software, H.-C.L. and Z.-Y.Y.; validation, H.-C.L. and W.-H.L.; visualization, H.-C.L. and Z.-Y.Y.; project administration, P.W.; funding acquisition, P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology of Taiwan under grant no. MOST 108-3116-F-168-001-CC2, no. MOST 109-2410–H-168-005 and the green energy technology research center on the featured areas research center program within the framework of the higher education sprout project from Ministry of Education (MOE) of Taiwan.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Problem formulation:
Symbol list.

c_1 and c_2	acceleration constants
C_i	route cost of path
d_{ij}	travel routing distance (hop count)
E	a set of edges e_{ij}
e_{ij}	edge from node x_i to node x_j
G	gravity vector
Lp	cost function
N	a set of network nodes
n_s	a set of nodes for attack sources
n_d	a set of victim
p_{ij}^n	the probability of a path from node i to node j for particle n
rand()	a random number in the range (0, 1)
QoS	quality of service
v_i^k	the velocity of particle i in k subswarm
v_o	initial speed of a particle
x_i^k	the position of particle i in k subswarm
Δx_i^k	the movement of Δt *for* particle i in k subswarm
$\Delta \tau_{ij}^k$	the movement of Δt for a particle in k subswarm (CFRS strategy)

References

1. Nguyen, A. Taiwan's Central Bank to Offer Banknote Exchange ahead of Lunar New Year. *Taiwan News*, 30 January 2018.
2. FS-ISAC. More than 100 Financial Services Firms Hit with DDoS Extortion Attacks. *Taiwan News*, 10 February 2021.
3. Catak, F. Two-layer malicious network flow detection system with sparse linear model based feature selection. *J. Natl. Sci. Found. Sri Lanka* **2018**, *46*, 601–612. [CrossRef]
4. Shi, Y.; Eberhart, R. A Modified Particle Swarm Optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
5. Angeline, P.J. Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Difference. In Proceedings of the 7th Annual Conference on Evolutionary Programming, International Conference on Evolutionary Computation, San Diego, CA, USA, 25–27 March 1998; pp. 69–73.
6. Lin, H.C.; Wang, P.; Lin, W.H. Implementation of an PSO-based security defense mechanism for tracing the sources of DDoS attacks. *Computers* **2019**, *8*, 88. [CrossRef]
7. Eberhart, R.C.; Shi, Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In Proceedings of the Congress on Evolutionary Computation, La Jolla, CA, USA, 16–19 July 2000; pp. 84–88.
8. Multi-Swarm Optimization. Available online: https://en.wikipedia.org/wiki/Multi-swarm_optimization (accessed on 25 February 2021).
9. Hendtlass, T. WoSP: A Multi-Optima Particle Swarm Algorithm. In Proceedings of the IEEE 2005 Congress on Evolutionary Computation, Edinburgh, UK, 2–5 September 2005; pp. 727–734.
10. Zhang, Q.W.; Zhan, F.F. Particle swarm optimization algorithm based on multi-subgroup harmony search. *J. Comput.* **2020**, *31*, 116–126.
11. Zhao, S.Z.; Liang, J.J.; Suganthan, P.N.; Tasgetiren, M.F. Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search for Large Scale Global Optimization. In Proceedings of the IEEE 2008 Congress on Evolutionary Computation, Hong Kong, China, 1–6 June 2008; pp. 3845–3852.
12. Xua, X.; Tang, Y.; Li, J.; Hua, C.; Guan, X. Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy. *Appl. Soft Comput.* **2015**, *29*, 169–183. [CrossRef]

13. Chen, S.; Montgomery, J. Selection Strategies for Initial Positions and Initial Velocities in Multi-Optima Particle Swarms. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; pp. 53–60.
14. Chen, S. Locust Swarms-A New Multi-Optima Search Technique. In Proceedings of the 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 18–21 May 2009; pp. 1745–1752.
15. Cuevas, E.; González, A.; Fausto, F.; Zaldívar, D.; Pérez-Cisneros, M. Multithreshold segmentation by using an algorithm based on the behavior of locust swarms. *Math. Probl. Eng.* **2015**, *2015*, 805357. [CrossRef]
16. A* Search Algorithm. Available online: https://en.wikipedia.org/wiki/A*_search_algorithm/ (accessed on 17 January 2020).

applied sciences

MDPI

Article

A Compute and Wait in PoW (CW-PoW) Consensus Algorithm for Preserving Energy Consumption

Mostefa Kara [1,*], Abdelkader Laouid [1], Muath AlShaikh [2], Mohammad Hammoudeh [3], Ahcene Bounceur [4,*], Reinhardt Euler [4], Abdelfattah Amamra [5] and Brahim Laouid [1]

[1] LIAP Laboratory, El Oued University, P.O. Box 789, El Oued 39000, Algeria; abdelkader-laouid@univ-eloued.dz (A.L.); laouid.bahi@gmail.com (B.L.)

[2] Computer Science Department, College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia; M.ALSHAIKH@seu.edu.sa

[3] School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester M1 5GD, UK; M.Hammoudeh@mmu.ac.uk

[4] Lab-STICC UMR CNRS, University of Western Brittany UBO, 6285 Brest, France; Reinhardt.Euler@univ-brest.fr

[5] Computer Science Department, California State Polytechnic University Pomona, 3801 W Temple Ave, Pomona, CA 91768, USA; aamamra@cpp.edu

* Correspondence: karamostefa@univ-eloued.dz (M.K.); Ahcene.Bounceur@univ-brest.fr (A.B.)

Abstract: Several trusted tasks use consensus algorithms to solve agreement challenges. Usually, consensus agreements are used to ensure data integrity and reliability in untrusted environments. In many distributed networking fields, the Proof of Work (PoW) consensus algorithm is commonly used. However, the standard PoW mechanism has two main limitations, where the first is the high power consumption and the second is the 51% attack vulnerability. In this paper, we look to improve the PoW consensus protocol by introducing several proof rounds. Any given consensus node should resolve the game of the current round $Round_i$ before participating in the next round $Round_{i+1}$. Any node that resolves the game of $Round_i$ can only pass to the next round if a predetermined number of solutions has been found by other nodes. The obtained evaluation results of this technique show significant improvements in terms of energy consumption and robustness against the 51% and Sybil attacks. By fixing the number of processes, we obtained an energy gain rate of 15.63% with five rounds and a gain rate of 19.91% with ten rounds.

Keywords: improved PoW consensus algorithms; blockchain; energy consumption; distributed systems; game competition

Citation: Kara, M.; Laouid, A.; AlShaikh, M.; Hammoudeh, M.; Bounceur, A.; Euler, R.; Amamra, A.; Laouid, B. A Compute and Wait in PoW (CW-PoW) Consensus Algorithm for Preserving Energy Consumption. *Appl. Sci.* **2021**, *11*, 6750. https://doi.org/10.3390/app11156750

Academic Editor: Gianluca Lax

Received: 30 May 2021
Accepted: 25 June 2021
Published: 22 July 2021

1. Introduction

Distributed computing is a computing field that studies distributed systems (DS) whose components are located on different networked computers spread over different geographies and which communicate by transmitting messages in order to achieve a common goal [1]. In this scenario and with the absence of a global clock, the event of failure of an independent component in the system must tolerate the failure of individual computers. Each computer has only a limited and incomplete view of the system. Various hardware and software architectures are used for DS. In peer-to-peer (P2P), there are no special machines that provide services or manage network resources and the responsibilities are evenly distributed among all machines called peers. Peers can serve as both clients and servers. In the context of DS, the Byzantine Generals Problem (BGP) is a distributed computational problem that was formalized by Leslie Lamport, Robert Shostak and Marshall Pease in 1982 [2]. The BGP is a metaphor that deals with the questioning of the reliability of transmissions and the integrity of the interlocutors. A set of elements working together must indeed manage possible failures between them. These failures will then consist of the presentation of erroneous or inconsistent information. The management of these faulty components is also called fault tolerance which leads us to talk about synchronous and asynchronous systems. Fischer, Lynch and Paterson (FLP) [3] have shown that consensus can not be reached in a synchronous environment if even a third of the processors are maliciously defective. In an asynchronous system, even with a single faulty process, it is not possible to guarantee that consensus will be reached (the system does not always end). FLP says that consensus will not always be reached, but not that it ever will be. This study concerns asynchronous systems, where all processors operate at unpredictable speeds [4]. Theoretically, a consensus can not always be achieved systematically asynchronously. But despite this result, it is possible to obtain satisfactory results in practice, as for instance by the non-perfect algorithms of Paxos Lamport [5] (in the context of obvious failures and no Byzantine faults). Lamport, Shostak and Pease showed in [2], via the Byzantine Generals Problem, that If f is the number of faulty processes, it takes at least $3f + 1$ processes (in total) for the consensus to be obtained. Under these conditions, the PoW technique has ensured the consensus perfectly, but its major problem is the enormous consumption of energy. In this paper, we propose a consensus protocol for an asynchronous environment. We minimize the energy consumption to ensure the synchronization by the application of several rounds of consensus, where in each round the nodes make a Proof-Wait, i.e., show the PoW then make a wait. The remainder of this paper is organised as follows: In Section 2, we will discuss the consensus problem. In Section 3, we provide an overview of the consensus protocol. In Section 4, we present our proposed protocol. Section 5 demonstrates the validity of our protocol. Section 6 shows the compute and wait implementation. In Section 7, we will discuss the hardness of the proposed cryptosystem. Finally, we conclude with Section 8.

2. Consensus Problem

A fundamental goal in distributed computing and multi-agent systems is to achieve overall system reliability in the presence of a number of faulty processes. This obviously requires the coordination of the correct processes in order to reach an agreement on a final decision. The processes must agree on a common value, where each process must provide a local value which is broadcast to all the other processes (or else, it shows a measure or a calculation). From all the proposed values, the processes must decide on a single common value such that either a leader process initiates the accord phase, or the accord phase is started at predetermined times. Many applications require consensus including blockchain, clock synchronization, cloud computing, opinion-forming, page-rank, smart grids, drone control, state estimation, load balancing and so on.

2.1. Conditions to be validated

The choice of the consensus algorithm is the main element. It determines the level of security and impact, and thus the following points must be achieved:

- Accord: The decided value is the same for all the correct processes.
- Integrity: A process decides at most once and there is no change in value choice.
- Validity: The value chosen by a process is one of the values proposed by the other processes.
- Termination: The decision phase takes place in a finite time (any correct process decides in a finite time).

2.2. Potential attacks

Choosing the wrong consensus algorithm can render the underlying system unusable and put all stored data at risk. The vulnerability of a consensus can expose the system to the following attacks:

- 51% attack: In the PoW algorithm, the domination can be achieved by controlling more than half of the total computation of the network (hash rate) [6]. The pool (a group of miners working together called a mining pool) would be able to add its own blocks to the blockchain or create a competing independent branch to which the main and legitimate branch will converge later. This type of attack notably allows the attacking

pool to be able to spend twice its own funds (double-spending attack) and reject transactions that it does not want to be included in the ledger.
- Sybil attack: One hostile node can conduct a Sybil attack by creating a large number of identities and using them to exert disproportionate influence and to defraud the system to break its trust and redundancy mechanism [7].
- DDoS attack: This attack aims to disrupt the normal functioning of the network by flooding the nodes with information or to lower the expected success outlook of a competing mining pool [8].

3. Consensus Protocol Overview

Bitcoin technology could refer to the most famous blockchain implementation that is created in 2008 by a person or group working under the pseudonym "Satoshi Nakamoto" [9]. In public cryptocurrencies and distributed ledger systems, the fundamental infrastructure of the blockchain is a peer-to-peer overlay network over the Internet [10]. Transactions represent the exchanges between users, and the recorded transactions are grouped together in blocks of size 1M at most. After recording recent transactions, a new block is generated and all transactions will be validated by miners, who will analyze the entire history of the blockchain. If the block is valid, it is time-stamped and integrated into the blockchain. The transactions' contents are then visible on the entire network. Once added to the chain, a block can not be changed or deleted, which guarantees the authenticity and security of the network. Each block in the chain is made up of the following elements: a collection of transactions, the hash (sum of transactions) used as an identifier, the hash of the previous block (except for the first block in the chain, called the genesis block) and the target (a measure of the amount of work that was required to produce the block). The main application of this technology is that of crypto-currencies such as Bitcoin [11]. Beyond its monetary aspect, this decentralized information storage technology could have multiple applications requiring secure exchanges without going through a centralizing body, or unfalsifiable traceability, such as applications based on smart contracts, applications allowing the exchange of all kinds of goods or services, means of improving their predictive systems known as oracles, the traceability of products in the food chain, etc. Each node of the network operates autonomously with respect to the set of rules to which it belongs, and this mechanism of identity management plays a main role in determining the organization of the nodes of a blockchain network.

From the system design perspective, a blockchain network contains four levels of implementation. These are the data and network organization protocols, the distributed consensus protocols, the autonomous organization framework based on intelligent contracts and the application (the interface) [12]. In each type of blockchain, several consensus algorithms are designed. One of the most famous algorithms is Proof of Work (PoW), whose concept was first introduced by Cynthia Dwork and Moni Naor in 1993 [13] and in which the authors have presented a computational technique to combat spam in particular and control access to a shared resource in general. The main idea is to require a user to calculate a moderately difficult but not insoluble function, in order to access the resource, thus avoiding frivolous use. Then the work should be difficult to do for the requester, but easily verifiable for a third party. In 1997, Adam Back implemented the idea with Hashcash, an algorithm to easily produce proofs of work using a hash function (especially SHA-256), and whose main use was electronic mail. The term 'proof of work' has been coined in 1999 by Markus Jakobsson and Ari Juels in their article Proofs of Work and Bread Pudding Protocols [14]. In Bitcoin, to validate a block, the miner had to build a draft of this block (including transactions and payload data), indicate the identifier of the previous block to make the link, and vary a number present in the header called the nonce. By varying this nonce (as well as other parameters in the block), the miner was able to try a gigantic number of possibilities so that the hash of the header produced a suitable result, i.e., a hash starting with a sufficient number of zeroes. Due to the high power consumption of PoW, the Proof of Stake (PoS) is positioned as an alternative. Peercoin was the first cryptocurrency to use PoS by Sunny King and Scott Nadal in 2012 [15]. PoS asks the user to prove the possession of a certain quantity of cyber money to pretend to validate blocks. To avoid centralization (the richest member would always have an advantage) and the Nothing at Stake attack, many alternatives exist for a move towards more comprehensive consensus mechanisms which use random allocation methods taking into account the age of the coin (as in the case of Peercoin) and depending on the velocity [16] used by the ReddCoin cryptocurrency. The variant that is often considered as one of the most balanced protocols between security, decentralization and network scalability is Delegated Proof of Stake used by the BitShares cryptocurrency [17]. Its selection is based on votes in which the block validator is randomly selected from a group of 101 delegates who have the highest stakes. Proof of Burn (PoB), or Proof of Destruction [18], is an algorithm very similar to PoS. In PoS, the participant sequesters a certain amount of cryptocurrency, which is a necessary collateral to participate in the validation of the network, but if he wishes to leave this network it is possible to recover his initial stake. What PoB and PoS have in common is the fact that block validators must invest their own coins in order to participate in the consensus mechanism. At PoB, this will involve destroying the coins that the participant provided to gain the right to validate network transactions. This system is similar to PoS in that the more coins it burns, the more likely it is to obtain the associated reward. Proof of Burn is offered as an alternative to the classic Proof of Work, but this young technique is criticized by some detractors who consider it a simple waste of tokens. It is the idea of destroying cryptocurrency in order to create it.

There are also many challenges that attempt to replace “Work” in PoW. For example, Proof of eXercise (PoX) in [19], where the challenge is to solve a real eXercise, a scientific computation problem based on a matrix. The authors chose matrix problems because matrices have interesting composability properties that help to solve the difficulty, collaborative verification and pool-mining, and also that matrix problems cover a wide range of useful real-world problems, being a primary abstraction for most scientific computing problems. The miner must solve the following equation:

$$X_1 \circ X_2 \circ \ldots \circ X_p = Y \tag{1}$$

where X_i and Y are matrices, $\circ$ is an operator, e.g., a product, a sum, etc. Another challenge proposed is Primecoin [20] which, as its name indicates, consists of finding prime numbers instead of finding the nonce.

Proof of Space [21] or Proofs of Capacity (PoC) is a protocol between a prover P and a verifier V which has two distinct phases. After an initialization phase, P is supposed to store data F of size N, while V contains only a few information. At any later time, V can initiate an execution phase of the proof, and at the end, V outputs reject or accept. The authors demanded that V be very efficient in both phases, while P is very efficient in the execution phase as long as it is stored and has random access to the data F. The simplest solution would be for the verifier V to generate a pseudo-random file F of 100 GB and send it to the prover P during an initialization phase. Later, V can ask P to return a few bits of F at random positions, making sure that V stores (at least a large part of) F. Unfortunately, with this solution, V still has to send a huge 100 GB file to P, which makes this approach virtually useless in practice. The PoC scheme which they proposed is based on graphs that are difficult to engrave. During the initialization phase, V sends the description of a hash function to P, which then labels the nodes of a graph that is difficult to engrave using this function. Here, the label of a node is calculated as the hash of the labels of its children. V then calculates a Merkle hash of all the labels and sends this value to P. In the execution phase of the proof, V simply asks P to open the labels corresponding to certain nodes chosen at random.

Proof of Space-Time (PoST) is another consensus algorithm closely related to PoC. PoST [22] differs from proof of capacity in that PoST allows network participants to prove that they have spent a “space-time” resource, meaning that they have allocated storage capacity to the network over a period of time. The authors called this ’Rational’ Proofs of

Space-Time because the true cost of storage is proportional to the product of storage capacity and the time that it used. The rational proof of financial interest in the network achieved by PoST addresses two problems with proof of capacity. The first, Arbitrary amortized cost: In a consensus system that doesn't account for time, participants can generate an arbitrary amount of PoC proofs by reusing the same storage space, and lowering their true cost. The second, Misaligned incentives: A rational participant in a PoC system will discard almost all stored data whenever computation costs less than the data storage do. This essentially turns PoC into a partial PoW system, which is potentially more resource-intensive.

An extension of Bitcoin's PoW via PoS is presented in Proof of Activity (PoA) by Bentov et al. [23]. Miners start with PoW and claim their reward. The difference is that the extracted blocks do not contain transactions. They are simply templates with header information and the mining reward address. Once this nearly blank block is mined, the system switches to PoS. The header information is used to select a random group of validators to sign the block. They are coin holders (stakeholders) and the greater the stake held by a validator, the more likely he or she will be selected to sign the new block. Once all the chosen validators have signed the block, it becomes an actual part of the blockchain. If the block remains unsigned by some of the chosen validators for a given time, it is rejected as incomplete and the next winning block is used. Validators are chosen again and this continues until a winning block is signed by all selected validators. The network costs are divided between the winning miner and the validators who signed the block. PoA is criticized that too much power is still needed to mine blocks during the PoW phase on one hand. On the other hand, coin accumulators are even more likely to make the signatory list and rack up more virtual currency rewards.

A random mining group selection to prevent 51% Attacks on Bitcoin is proposed in [24]. The authors divide miners into several groups. Each peer node determines its mining group using the Hg $(\cdot)$ hash function and its wallet address. Additionally, once a block is created, its hash value is used with Hg $(\cdot)$ to determine which mining group is supposed to find the next block. Only even nodes belonging to the mining group are allowed to mine the next block and to compete with each other. Once a block is propagated over the P2P network, other nodes can check if the block was generated by the correct mining group by comparing the hash value of the previous block in the header of the block with the address of block creators. Here, although there may be an attacker with more than half of the total hash power, the chances of a successful double-spend attack can be greatly reduced by increasing the number of mining groups as the mining groups are chosen at random. In addition, the computational power required for block mining is effectively reduced by 1/(number of groups) because even nodes not belonging to the selected group do not participate in PoW and the difficulty level can be lowered due to the smaller number of competing miners in each group. The authors show that if the number of groups is greater than or equal to two, the probability of the attacker of finding the next block is less than 50%.

4. Proposed Protocol

There are three types of blockchain, private, public and hybrid. A private blockchain (permissioned) operates in a restrictive environment, i.e., a closed network. In an authorized blockchain that is under the control of an entity, only authorized nodes with a revealed identity are allowed to enable basic functionalities such as consensus participation or data propagation [25]. Comparatively, in a public blockchain (permission-less/open access), if the node has a valid pseudonym (account address), it can freely join the network and activate any available network functionalities such as sending, receiving and validating transactions and blocks according to common rules. Therefore, there is usually such a blockchain network instance on a global scale that is subject to public governance. Specifically, anyone can participate in the blockchain consensus, although a person's voting power is generally proportional to its possession of network resources, such as computing power, wealth token and storage space [26]. A hybrid blockchain (consortium or federated) is a creative approach to solving the needs of organizations which have a need for public and private blockchain functionality. Some aspects of organizations are made public, while others remain private. The consensus algorithms in a consortium blockchain are controlled by the predefined nodes. It is not open to the popular masses, it still has a decentralized character and there is not a single centralized force that controls it. Therefore, it offers all the functionalities of a private blockchain, including transparency, confidentiality, and efficiency, without a single party having to consolidate power. In this paper, we are concerned with the second type only, which is the public blockchain.

As shown in Figure 1, we divided the overall operation to reach consensus into several rounds (Supplementary Materials). At the start, a node launches the first round and looks for a solution for its own block, like the basic PoW algorithm, but with a much lower degree of difficulty than what is currently applied. Where the difficulty was X and the number of rounds was 1, in the proposed algorithm, the difficulty is $X' < X$ and the number of rounds is $Y > 1$. Once the solution is found, the node shares it and checks whether there are nine (9) other solutions found in the network for this round (for example, in a scenario of 10 solutions to find). If so, this node has the right to participate in the next round and to restart the PoW. If not, i.e., there are not yet nine (9) solutions found by other nodes, this candidate node will wait until it receives the remaining nine (9) solutions. In the last round, the first node that will find the solution will be the miner, so that in the last round the protocol looks for a single solution. In the original PoW, the proof consists of finding the nonce according to the inequality: $Hash(Block + Nonce) < Target$. In the proposed protocol, the proof of round i is according to the following inequality:

$$Hash(Block + IDRound_{i-1} + Nonce) < Target \tag{2}$$

Initially, the identifier (ID) Round is equal to 1. After that, the *ID Round* is equal to the sum of nonces found in the previous round. Therefore, in each round, there is a new ID so that the nodes will work on it. If we have ten solutions to find in each round, we will obtain the following equation:

$$IDRound_k = \sum_{i=1}^{10} (nonce_i \; of \; round_{k-1}) \tag{3}$$

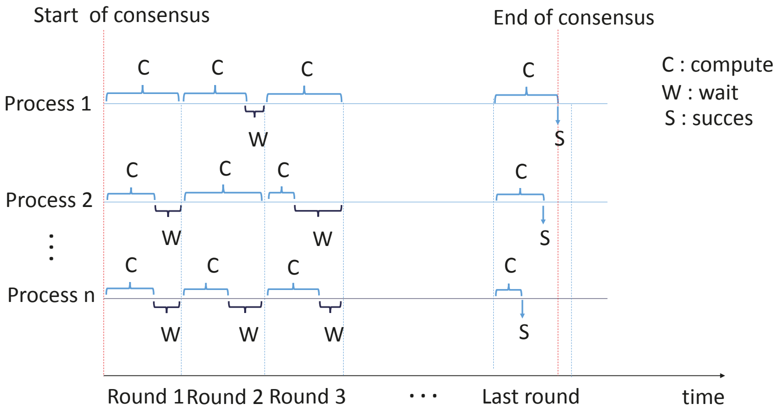


Figure 1. A Compute and Wait Consensus PoW Algorithm.

Let NbrR be the number of rounds and NbrS the number of solutions to find in each round. If two nodes succeed in finding the solution in the last round, then the other nodes will receive two solutions. In this case, the nodes must calculate the standard deviation of the solutions found in all the rounds for each of these two winners. The miner is the node with the smallest standard deviation. There is another parameter that we can introduce here, which is the consensus state. For the moment, in which round do the processes work? Assuming that a process decides to leave competition if the other processes exceed it by 5 rounds (or 4, for example), this will minimize the energy to be consumed. For a process, if the other competitors overtake him by two rounds, there is little hope of catching them. This process will lose energy unnecessarily if it continues the calculation.

5. Protocol Demonstration

The four conditions of a consensus are: accord, integrity, validity and termination. In fact, we see that there is a fifth condition that must be satisfied in every consensus to be perfect. This condition is equality of opportunity (no domination). In the following, We explain how our protocol verifies these five conditions.

1. Accord: In the final round, in the event that there is only one winner, all the nodes will agree on him. In the case where there is more than one winner, for each winner, we will calculate the standard deviation of his solutions found during all the rounds and we will choose the minimum of these standard deviations. Mathematically, this value is unique. In case there are two branches of the blockchain (fork problem), this problem is solved in basic PoW by the golden rule, where the nodes will choose the longest chain. Finally, the decided value is the same for all the correct nodes and in all cases.
2. Integrity: Once the P_x process obtains a valid nonce $Hash\ (Block_i + Nonce_1) < Target$, the process broadcasts it over the network with a specific date (timestamp). Of course, this nonce is concerned with a specific block (B_{i+1}) and Nonce (N_1), i.e., $P_x(B_{i+1}, N_1)$. After having obtained a second nonce N_2, the process will broadcast it ($P_x(B_{i+1}, N_2)$), and we notice here that there are two nonces N_1 and N_2 for the same block B_{i+1}. In this case, there are several actions we can take, including to choose the smallest nonce. Eventually, each process will participate with at most one value, so we have fulfilled the condition of integrity.
3. Validity: For each solution received, the node will check its validity. The miner at the end is a network node, having a unique public address and resolving the PoW in all rounds.
4. Termination: Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash [6]. PoW resolution is estimated at 10 min, and these 10 min are more than enough for the propagation in the network, and the candidate block will have been added to the blockchain. In the proposed technique, the difficulty is minimized and the number of rounds is increased. To ensure the termination in a finite time, we just need to balance between the two factors difficulty and rounds. So, if the PoW has a termination, the proposed protocol also has a termination.

6. Implementation

In our experiment, we simulated each node by a process, where we implemented multi-process programming. In several tests, we created a different number of processes, each of them starting with the creation of a random number $blockID \leftarrow random()$, which is considered as the ID of the candidate block that the process is working on. After that, each process follows the execution of the proposed algorithm (Algorithm 1). Using five rounds and ten processes, the test took about 5 min, and we got the result shown in Figure 2. The execution at University of El Oued was defined in Python using a mainframe made up of 32 dual-processor nodes of 10 cores each. The intensive computing unit has a management node with the following specifications: Processor: Intel Xeon (R) E5-2660 v3 @ 2.60 GHz x 20, memory: 64 GB of RAM, disk: 2 TB HDD, OS: RedHat Enterprise Linux Server 7.2, OS type: 64 bit. The computing unit has 32 nodes. Each node has the following characteristics: 10 physical cores, storage capacity: 500 GB HDD and available memory: 64 GB. The Proposed technique can be defined by the following algorithm:

Algorithm 1 Consensus algorithm

```
Require: $NbrS, NbrR$
Ensure: $solution$

procedure Cons($NbrS, NbrR$)
    $sols \leftarrow 0$
    $roundID \leftarrow 0$
    $currentRound \leftarrow 1$
    $q \leftarrow Queue()$
    $blockID \leftarrow random()$
    while $currentRound <= NbrR$ do
        while $solution\ not\ found$ do
            $search\ for\ solution$
        end while
        $sols \leftarrow sols + 1$
        if $currentRound\ is\ NbrR$ then
            $print(I\ am\ the\ winner)\ ,\ Exit()$
        end if
        $Broadcast(currentRound,\ solution)$
        $put\ on(q,\ solution)$
        while $sols < NbrS$ do
            $Nothing$
        end while
        $sols \leftarrow 0$
        $currentRound \leftarrow currentRound + 1$
        $RoundID \leftarrow \sum_{i=1}^{NbrS} q(i)$
        $Broadcast(currentRound,\ RoundID)$
    end while
end procedure
```

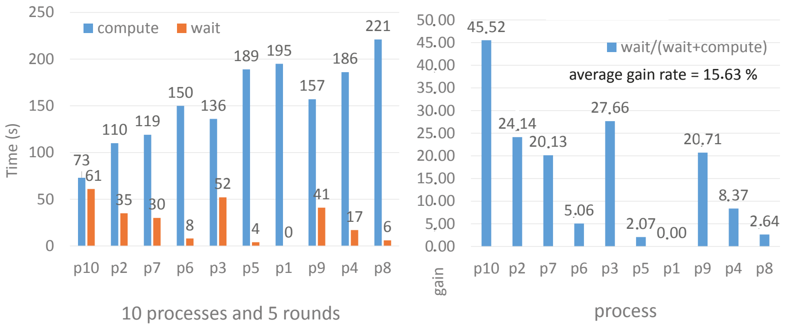


Figure 2. Compute and wait example execution.

We repeated the experiment tens of times in each scenario (each input variation). For example, in Figure 3 with the first scenario where the number of processes is equal to 5 and the number of rounds equal to 5, the gain was 12.07. We repeated this test several times and we obtained different values (between 11 and 14), but we took the most frequent value which was 12.07. We did this at each input change (NbrR, NbrP = (5, 5); (5, 10); (5, 15); (5, 20). We did not take the average of measurements, but we considered the most frequent value.

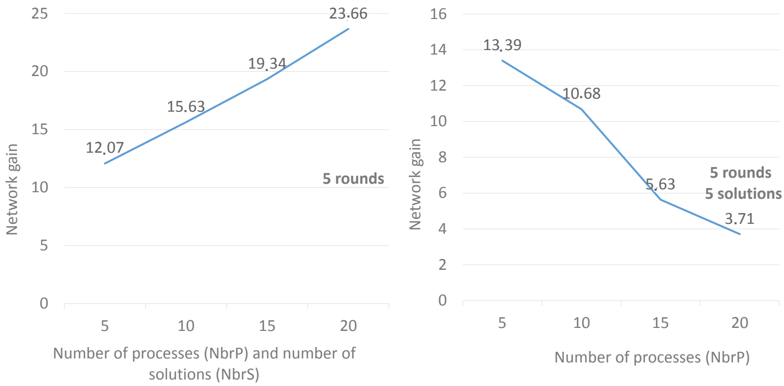


Figure 3. The influence of NbrP for fixed NbrR.

The formal definition of gain: In each round, the process searches for a solution (compute time). Then the process will wait until it receives the latest solution. We will consider this wait time as a gain, and we can formulate the gain rate as follows: *gain rate = wait time/total execution time*. If the total execution time is the wait and compute time, we obtain:

$$gain\ rate = wait\ time/(wait\ time + compute\ time) \tag{4}$$

We let $C_{i,x}$ (respectively, $W_{i,x}$) be the compute (respectively, wait) time of the process i in round x. The gain of the process i can be formulated as follows:

$$gain_i = \sum_{j=1}^{NbrR} (W_{i,j}/(W_{i,j} + C_{i,j})) \tag{5}$$

The average gain rate (network gain rate) is:

$$Gain = \sum_{i=1}^{NbrP} (gain_i/NbrP) \tag{6}$$

In the scenario shown in Figure 2, we have set the number of rounds $NbrR = 5$, $NbrP = NbrS = 10$, where NbrP (respectively, NbrS) is the number of processes (respectively, of solutions). We do not put different values between $NbrP$ and $NbrS$ in order to study the pure effect of the number of rounds. We notice that there is a real compute and wait in each round for most of the processes. Process 10 is the fastest because it has the minimum execution time and the maximum waiting time, that is why it has the highest gain rate (45.52%). On the other hand, process 1 ($p1$) is the heaviest process, because $p1$ has the highest execution and the lowest waiting time, and thus the lowest gain rate. For $p1$, the gain rate = wait/(wait + execution) = 0/(0 + 195) = 0. We explain that $p1$ is the last process to find the solution, it wakes up the others, and then starts directly the next round without making any wait. Even though $p1$ didn't wait, its computing time is less than the one of $p8$. We explain this by two possibilities. The first is given when $p8$ started, the second possibility when $p1$ quickly found the solution in the last round (there is no wait after the last round) and $p8$ took a long time in the last round. In general, the average gain of the network is 15.63%.

Compute and wait implementation: In the original PoW, each process will continue the calculation for 10 min (until one of the processes finds the hash). In the proposed protocol there are two types of processes, winning processes, which are processes that participate in all rounds (noting that the final winner is the process that has the minimum standard deviation of his solutions). The other type are those processes that leave the competition after such rounds (for example, when the number of solutions of the next round is equal to NbrS). In these two types, no process does the calculation during 10 min, there is a proof-wait (first type) or a proof-abandon (second type). So, there are two main factors that must be handled, the number of rounds $NbrR$ and the number of solutions to be found in each round $NbrS$.

Lemma 1. *If* $NbrR > 1$ *then* $Gain > 0$

Proof. Based on the randomness of the hash function (Block + Nonce) < Target and on the arbitrary speed of each process, we have:

$$C_{i,x} \neq C_{j,x},\ \forall i,j \in \{1,...,NbrP\},\ \forall x \in \{1,...,NbrR\} \tag{7}$$

where $NbrP$ is the number of processes and $0 < x < NbrR$. According to Equation (7), $|C_{i,x} - C_{j,x}| > 0$, so either $W_{i,x} > 0$ or $W_{j,x} > 0$, which implies that $W_{i,x}/(W_{i,x} + C_{i,x}) > 0$ or $W_{i,x}/(W_{j,x} + C_{j,x}) > 0$. We obtain $Gain > 0$. □

Lemma 2. *If* $NbrP$ *increases then* $Gain$ *increases when* $NbrP = NbrS$

Proof. We let the probability of finding the solution by 5 (respectively, 10) processes be P_5 (respectively, P_{10}), we have $P_{10} > P_5$. Let T_p be the compute time to find the solution with a probability p. So, $T_{p_{10}} < T_{p_5}$ implies that the wait time $W_{p_{10}} > W_{p_5}$. According to Equation (4), $gain_{10processes} > gain_{5processes}$. □

Discussion: In the first scenario shown by Figure 3, we fixed NbrR to 5, and we made several tests, during which we increased the number of processes (NbrP = 5, 10, 15, 20) to study the effect of this increase and its influence on the gain. These tests have shown that the more many processes do compute and wait, the more the gain increases. We explain this by the following example: in the round x, if first process p_i finds a solution after 4 min, we suppose that p_i will wait $t_1 = 1$ minute to start the next round $x + 1$, so the gain is $1/(1 + 4) = 20\%$. With $NbrP = 10$, the first process p_j will find a solution in 3 min, p_j will wait for $t2 > t1$ ($t2 = 2$ min), so the gain is 2 / (2 + 4) = 30%.

In the second curve (curve on the right in Figure 3), where the NbrS is constant, we observe that the gain decreases, because the number of processes (which are waiting) decreases compared to the total number of processes. On the other hand, the number of processes that do not wait is increasing, which implies a decrease in the average gain in the network.

Lemma 3. *While* $NbrR < upper\ bound$*, if* $NbrR$ *increases then Gain increases.*

Proof. Let w be the wait time, c the compute time, and e the execution time, so $e = w + c$. The gain $g = \frac{w}{e}$, and if a process does little computation and a lot of wait, then it will have a high gain rate. We know that there is no wait time in the last round, therefore, if $NbrR = 2$, then $g = \frac{w}{e_1+e_2}$ where w is the wait time of round 1, e_1 the compute time of round 1, and e_2 the compute time of round 2. Let $e_1 + e_2 = 2 \times e$. If $NbrR = 3$, then $g = \frac{2\times w}{3\times e}$, if $NbrR = 4$, then $g = \frac{3\times w}{4\times e} \ldots$, if $NbrR = \alpha$, then $g = \frac{\alpha\times w}{(\alpha+1)\times e}$. The effect of the wait time absence in the last round decreases if we increase $NbrR$ and the gain g converges to $\frac{w}{e}$ because $\frac{\alpha}{(\alpha+1)}$ converges to 1. □

Discussion: In the second scenario illustrated by Figure 4, we fixed $NbrP$ to 10, and we made several tests, in each of them increasing the number of rounds ($NbrR = 5, 10, 15, 20$) to study the effect of this increase and its influence on the gain. These tests have shown that the more $NbrR$ increases, the more the gain increases but not absolutely. Rather, this increase converges to an upper bound related to the number of processes. Then the gain decreases because the difficulty is also reduced (NbrR increases then difficulty decreases).

Therefore, the processes will end almost at the same time because the solution is easy to find.

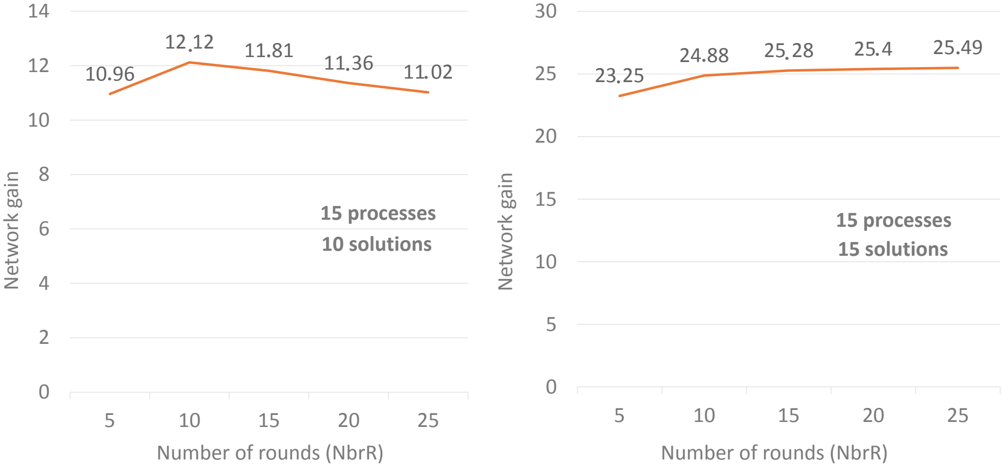


Figure 4. The influence of NbrR, case A.

Lemma 4. *If $NbrR$ increases then $Gain(G_{nw})$ converges to a number G*

Proof. The average gain of the network $(G_{nw}) = (g_1 + g_1 + \dots g_{NbrR})/NbrR$ where g_i is the gain of round i. We have g_{NbrR} (gain of the last round) equal to 0 because there is a calculation and there is no wait. g_i depends on moments $t_{i,j}$ at which solutions are found ($t_{i,1}$ is the instant where the first solution of round i is found, $t_{i,NbrR}$ is the instant at which the last solution of round i is found). In general, these times are random in each round, therefore, $g_1 \approx g_2 \approx \dots g_{NbrR-1}$ and then $G_{nw} = (NbrR - 1) \times g_1/NbrR$ □

We obtain:

$$G_{nw} = g_1 - (g_1/NbrR) \tag{8}$$

We observe that if $NbrR = 1$ then $G_{nw} = 0$ and if $NbrR\ increases$ then $G_{nw} \approx g_1 = G$

Discussion: Figure 5 shows how the gain converges to a constant number G when the number of rounds increases (NbrR = 5, 10, 15, 20, 25). Usually, G depends on the number of processes and the number of solutions. In Figure 5, we have fixed the number of processes and the number of solutions. In Figure 5-right, (NbrP, NbrS) = (10, 5) and we have set up several scenarios (NbrR = 5, … until NbrR = 200). We noticed that starting from NbrR = 20 the gain remains greater than 8 and less than 9. In the similar way, in Figure 5-left, the gain remains close to 19.5 whenever NbrR > 20 (20 ≦ NbrR ≦ 200). This is why we said that the gain converges to a constant number G when the number of rounds increases. $G \approx 8.5$ with (NbrP, NbrS) = (10, 5) and NbrR ≧ 20; $G \approx 19.5$ with (NbrP, NbrS) = (10, 10) and NbrR ≧ 20.

The number of solutions is a very important factor because it designates the number of processes that will wait; therefore, it controls the gain. Figure 6 shows the increase in gain as NbrS increases. On the other hand, we cannot introduce a number of solutions without having a multi-round environment. The importance of NbrS directs us towards private blockchains; In which we can talk about NbrS compared to the total number of processes. In another philosophy, we may consider that the node leaves the competition if NbrS of the next round is saturated. In this case, we will increase the gain to the maximum (Table 1). The downside of this philosophy is that after a round j in which the number of processes is NbrS and one of these processes breaks down, the rest of the competitors (NbrS-1) cannot continue because NbrS (in round $j + 1$) will never be reached.

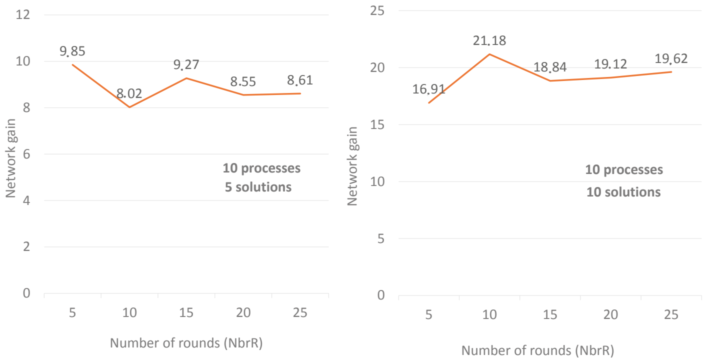


Figure 5. The influence of NbrR, case B.

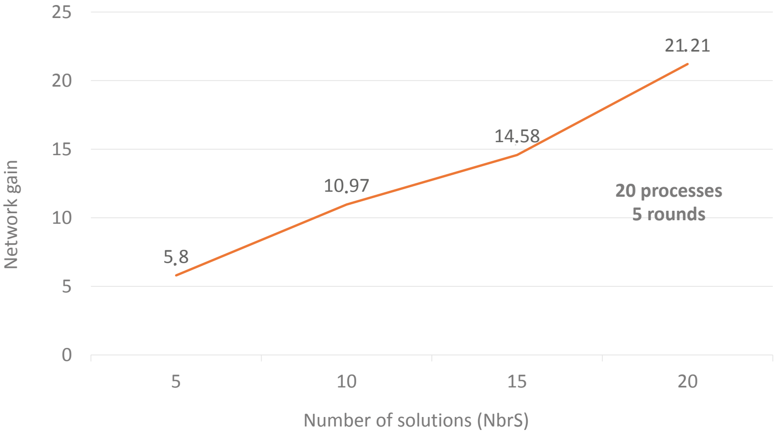


Figure 6. The influence of NbrS.

Table 1. The gain rate with leaving competition.

NbrR	NbrS	NbrP	Gain Rate
5	5	10	55
5	5	20	60
5	5	30	70
5	5	100	90

7. Analysis

To increase the level of security, PoW is based on the degree of difficulty. However, it remains weak against 51% attacks [27]. Regarding domination, when the difficulty is equal to D, there will be only one candidate (N_1) who has the greatest computing power. We have already done a test and reduced the difficulty to 50% to obtain 10 candidates, so the probability that N_1 wins is 10% instead of 100%, because no process controls the standard deviation of its solutions. Therefore, the dominance has been reduced.

As for the 51% attack, when dominance is reduced, the 51% attack will also be reduced. On the other hand, the pool can not work freely to find a single solution, it has to go through several rounds. After each round, the node must share the identifier of this round (id_{round}) which is calculated from the solutions trounced in the previous round, from which the protocol implies to introduce this id_{round} in the next proof of round (Equation (3)). These steps will slow down the operation of producing another longer chain (branch) for use in double-spending.

Speaking of the Sybil attack, it is in principle impracticable unless the attacker has more than 50% of the network's computing resources. Also, the consensus mechanism does not prevent an attacker from disrupting the network by creating a number of malicious nodes (false identities). The proposed protocol combines two techniques, multi-rounds and standard deviation. In comparison to the basic PoW, it is difficult for these two techniques and for a false identity to be the final winner. These false identities must create a chain longer than the chain known in the network. The first technique (multi-rounds) will slow down their work, the second (standard deviation) will decrease their chances.

8. Conclusions and Future Work

In this work, we have proposed an effective and applicable consensus algorithm, and shown that, in blockchain or in any setting where we need an agreement, it is adaptable. We have studied its validity according to the four known conditions of the agreement. We have shown the energy gain achieved by this protocol to reach a drop of 15.63% with five rounds and to reach a drop of 19.91% with ten rounds. We have set up several scenarios establishing in each one several tests, with the manipulation of three factors (number of rounds, number of processes and number of solutions in each round). We have studied separately the influence of each factor on the energy consumed, and also, the influence of introducing these factors in comparison to the basic PoW. We have seen the robustness of the algorithm against the most famous attacks (51% and Sybil attack). In the future, we intend to implement this algorithm in other sectors such as healthcare, for instance.

Supplementary Materials: Compute and Wait Proof o Work (CW-PoW) Video is available online at https://mmutube.mmu.ac.uk/media/POW_Demonstration/1_zrjr8fqq.

Author Contributions: Formal analysis, A.B.; Methodology, M.K., A.L. and R.E.; Software, M.H.; Supervision, A.A.; Validation, M.A. and B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Garcia-Molina, H. Elections in a distributed computing system. *IEEE Trans. Comput.* **1982**, *C-31*, 48–59. [CrossRef]
2. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. In *Concurrency: The Works of Leslie Lamport*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 203–226.
3. Fischer, M.J.; Lynch, N.A.; Paterson, M.S. Impossibility of distributed consensus with one faulty process. *J. ACM (JACM)* **1985**, *32*, 374–382. [CrossRef]
4. Turek, J.; Shasha, D. The many faces of consensus in distributed systems. *Computer* **1992**, *25*, 8–17. [CrossRef]
5. Lamport, L. The part-time parliament. In *Concurrency: The Works of Leslie Lamport*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 277–317.
6. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2019; pp. 1–9. Available online: www.bitcoin.org (accessed on 20 May 2021).
7. Zhang, S.; Lee, J.H. Double-spending with a sybil attack in the bitcoin decentralized network. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5715–5722. [CrossRef]
8. Johnson, B.; Laszka, A.; Grossklags, J.; Vasek, M.; Moore, T. Game-theoretic analysis of DDoS attacks against Bitcoin mining pools. In Proceedings of the International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 3–7 March 2014; pp. 72–86.

9. Nakamoto, S.A Peer-to-Peer Electronic Cash System. Bitcoin—URL. 2008; Volume 4. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 20 May 2021).
10. Rathi, V.K.; Chaudhary, V.; Rajput, N.K.; Ahuja, B.; Jaiswal, A.K.; Gupta, D.; Elhoseny, M.; Hammoudeh, M. A Blockchain-Enabled Multi Domain Edge Computing Orchestrator. *IEEE Internet Things Mag.* **2020**, *3*, 30–36. [CrossRef]
11. Mudassir, M.; Bennbaia, S.; Unal, D.; Hammoudeh, M. Time-series forecasting of Bitcoin prices using high-dimensional features: A machine learning approach. *Neural Comput. Appl.* **2020**, 1–15. [CrossRef] [PubMed]
12. Wang, W.; Hoang, D.T.; Hu, P.; Xiong, Z.; Niyato, D.; Wang, P.; Wen, Y.; Kim, D.I. A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access* **2019**, *7*, 22328–22370. [CrossRef]
13. Dwork, C.; Naor, M. Pricing via processing or combatting junk mail. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 1992; pp. 139–147.
14. Jakobsson, M.; Juels, A. Proofs of work and bread pudding protocols. In *Secure Information Networks*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 258–272.
15. King, S.; Nadal, S. *Ppcoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake*; Self-Published Paper, August; Peace Consortium of Defense Academies and Security Studies Institutes: Garmisch-Partenkirchen, Germany, 2012; Volume 19, pp. 1–6.
16. Ren, L. Proof of Stake Velocity: Building the Social Currency of the Digital Age. Self-Published White Paper. 2014; pp. 1–13. Available online: www.reddcoin.com (accessed on 20 May 2021).
17. Larimer, D. Delegated Proof-of-Stake (dpos). Bitshare Whitepaper. 2014. Available online: http://107.170.30.182/security/delegated-proof-of-stake.php (accessed on 20 May 2021).
18. Karantias, K.; Kiayias, A.; Zindros, D. Proof-of-burn. In Proceedings of the International Conference on Financial Cryptography and Data Security, Kota Kinabalu, Malaysia, 10–14 February 2020; pp. 523–540.
19. Shoker, A. Sustainable blockchain through proof of exercise. In Proceedings of the IEEE 16th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 30 October–1 November 2017; pp. 1–9.
20. King, S. Primecoin: Cryptocurrency with prime number proof-of-work. *citeseerx.ist.psu.edu* **2013**, *1*, 1–6.
21. Dziembowski, S.; Faust, S.; Kolmogorov, V.; Pietrzak, K. Proofs of space. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2015; pp. 585–605.
22. Moran, T.; Orlov, I. Simple proofs of space-time and rational proofs of storage. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 2019; pp. 381–409.
23. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM Sigmetrics Perform. Eval. Rev.* **2014**, *42*, 34–37. [CrossRef]
24. Bae, J.; Lim, H. Random mining group selection to prevent 51% attacks on bitcoin. In Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Luxembourg, 25–28 June 2018; pp. 81–82.
25. Xiao, Y.; Zhang, N.; Lou, W.; Hou, Y.T. A survey of distributed consensus protocols for blockchain networks. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1432–1465. [CrossRef]
26. Unal, D.; Hammoudeh, M.; Kiraz, M.S. Policy specification and verification for blockchain and smart contracts in 5G networks. *ICT Express.* **2020**, *6*, 43–47. [CrossRef]
27. Shi, N. A new proof-of-work mechanism for bitcoin. *Financ. Innov.* **2016**, *2*, 1–8. [CrossRef]

symmetry

Article

Microgrid Operations Planning Based on Improving the Flying Sparrow Search Algorithm

Trong-The Nguyen [1,2], Truong-Giang Ngo [3,*], Thi-Kien Dao [1,*] and Thi-Thanh-Tan Nguyen [4]

[1] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350014, China; jvnthe@gmail.com
[2] Multimedia Communications Lab., VNUHCM-University of Information Technology, Ho Chi Minh City 700000, Vietnam
[3] Faculty of Computer Science and Engineering, Thuyloi University, 175 Tay Son, Dong Da, Hanoi 116705, Vietnam
[4] Faculty of Information Technology, Electric Power University, Hanoi 100000, Vietnam; tanntt@epu.edu.vn
* Correspondence: giangnt@tlu.edu.vn (T.-G.N.); jvnkien@gmail.com (T.-K.D.)

Abstract: Microgrid operations planning is crucial for emerging energy microgrids to enhance the share of clean energy power generation and ensure a safe symmetry power grid among distributed natural power sources and stable functioning of the entire power system. This paper suggests a new improved version (namely, ESSA) of the sparrow search algorithm (SSA) based on an elite reverse learning strategy and firefly algorithm (FA) mutation strategy for the power microgrid optimal operations planning. Scheduling cycles of the microgrid with a distributed power source's optimal output and total operation cost is modeled based on variables, e.g., environmental costs, electricity interaction, investment depreciation, and maintenance system, to establish grid multi-objective economic optimization. Compared with other literature methods, such as Genetic algorithm (GA), Particle swarm optimization (PSO), Firefly algorithm (FA), Bat algorithm (BA), Grey wolf optimization (GWO), and SSA show that the proposed plan offers higher performance and feasibility in solving microgrid operations planning issues.

Keywords: microgrid; distributed power supply; enhanced sparrow search algorithm; economical operation

Citation: Nguyen, T.-T.; Ngo, T.-G.; Dao, T.-K.; Nguyen, T.-T.-T. Microgrid Operations Planning Based on Improving the Flying Sparrow Search Algorithm. *Symmetry* **2022**, *14*, 168. https://doi.org/10.3390/sym14010168

Academic Editors: Youcef Gheraibia and Yunlong Shang

Received: 6 November 2021
Accepted: 13 December 2021
Published: 15 January 2022

1. Introduction

Emerged microgrid technology will enhance the share of clean energy power generation, ensuring safe and stable functioning of the entire power system, maximizing the use of scattered power sources, and coordinating and optimizing control [1]. The microgrids can be considered an area's internal power system and an internal load that has power generation, transmission, and distribution capabilities to fulfill the dynamic load and power quality [2]. As a result, several countries have used it as a research focal point for power growth in the coming years [3]. Microgrid technology may not only improve the power supply quality in remote regions, e.g., mountainous locations, islands, and other areas [4], but it can also efficiently prevent large-scale power outages caused by accidents and disasters [5]. Furthermore, as clean energy power-generating technologies become more widely integrated, the composition and structure of power sources will become more complicated and diversified, causing challenges with reactive power balance and power quality across the entire power generation system [3].

Optimal operations planning for power supply systems would bring economic, safety, reliability, and low pollution, which is considered a microsystem [4,6] group composed of distributed a power supply, load, energy storage, control device system, and control device [7,8]. The traditional methods for a large power grid show a burden that suffers from greater computational complexity in its optimization [9,10]. Fortunately, the metaheuristic algorithm effectively deals with these issues of traditional optimization approaches for the optimal operation of large power grid systems [11,12].

Most metaheuristic algorithms are inspired by physical phenomena or natural species [13], e.g., the Firefly algorithm (FA) [14] and Sparrow search algorithm (SSA) [15] are from the behavior of finding prey and escaping threat or avoiding enemies. There are typical metaheuristic algorithms, such as Genetic algorithm (GA) [16], Particle swarm optimization (PSO) [17], Harmony search algorithm (HS) [18], Ant colony algorithm (ACO) [19], Bat algorithm (BA) [20], Grey wolf optimization (GWO) [21], and other evolutionary algorithms [22]. Scholars in several engineering domains are interested in metaheuristic algorithms because of the simple parameters, ease of understanding, and implementing procedures. The challenge is tackled by targeting optimization using a program that simulates a natural process iteratively.

A metaheuristic algorithm finds the best solution by allowing it to evolve naturally among populations of possible solutions. Because of the high chance of a successful search and the fast convergence speed, metaheuristic algorithms are increasingly used to solve complex engineering, healthcare, finance, and military issues [13]. The SSA [15] is a recent and excellent metaheuristic algorithm; it still faces the optimum local issue or can drop in the trap of the optimum local when dealing with complicated problems like the microgrid scheduling cycles problem.

This paper suggests a new enhanced SSA (ESSA) based on an elite reverse learning strategy and FA mutation strategies for optimal operations planning of microgrid schedule cycles with a distributed power source's optimal outputs to enhance the share of clean energy power generation and ensure a safe symmetry power grid among distributed natural power sources and stable functioning of the entire power system. The microgrid scheduling cycle running of the economy as the target and considering the differences of impact parameters, e.g., load demand, season, timesharing, and electricity price, is given as each unit's mathematical model in a power microgrid.

The contributions are highlighted as follows:

- A new optimization method (called ESSA) is proposed based on the SSA by applying elite reverse learning and FA's mutation strategies.
- The proposed ESSA's performance is evaluated by testing the selected benchmark functions.
- The microgrid scheduling cycle running of the power microgrid is mathematically modeled as the economy as the objective function for the optimization planning problem.
- The new proposed ESSA approach is applied to solve the microgrid scheduling cycle with the power source planning's optimal output and total operation cost.

The remaining paper is structured as follows. Section 2 presents the model description of the power grid problem. Section 3 proposes the ESSA and tests its validation and performance. Section 4 presents a case simulation on the power microgrid system to verify the effectiveness of the proposed ESSA. Section 5 concludes.

2. A Microgrid Optimizing Model

The requirement of the microgrid fulfils the dynamic load and power quality, efficiently preventing power outages that improve the power supply quality in isolated or remote regions [4]. A typical microgrid structure is composed of distributed power sources, such as mainly including wind turbines (WTs), photovoltaic (PV), energy storage systems (ESs), microturbines (MTs), fuel cells (FCs), and various parts of a load set, which is connected to the distribution network through the point of common coupling (PCC) voltage control of a grid [23]. In isolated island operation, the distributed power supply and generator set the microgrid load demands [24]. Figure 1 illustrates a typical microgrid structure schematic of distributed power sources.

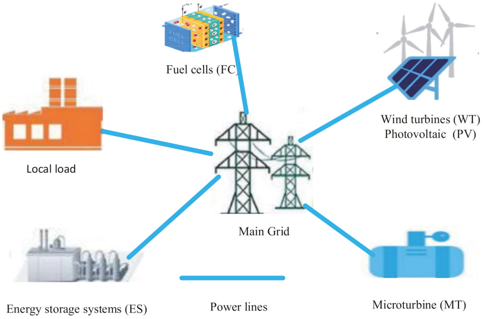


Figure 1. A typical microgrid structure schematic of distributed power sources.

WT power generation has characteristics that the actual wind speed mainly determines the wind turbines' output. The characteristics output values have several factors of the wind turbine's output power related to the wind speed as a piecewise function modeled following the output power model:

$$P_{wt}(v) = \begin{cases} 0, & 0 \leq v < v_{ci} \\ A + Bv + Cv^2, & v_{ci} \leq v \leq v_r \\ P_x, & v_r \leq v \leq v_{co} \\ 0, & v_{co} < v \end{cases} \tag{1}$$

where v_{ci} is the cut wind speed; v_r is the rated wind speed; v_{co} is the cutting wind speed; P_x is the rated power of the fan; and A, B, C are the power characteristic curve parameters of the *WT*.

PV power supply as the output power supply can be expressed as follows:

$$P_{pv} = P_{rpv}\frac{I_F(t)}{800}[1 + \lambda_t(T(t) - T_{st})]\eta_{pv} \tag{2}$$

$$T(t)a = 25.5 + \frac{I_F(t)}{800}(T_{sc} - 20) \tag{3}$$

where P_{pv} is the output power value of the photovoltaic power supply; P_{rpv} is the rated power value of the photovoltaic system; $I_F(t)$ is the actual illumination intensity at the sampling point t in the optimized operation; λ_t is the power-temperature coefficient; $T(t)$ is the ambient temperature value of the photovoltaic cell sampling point at time t; T_{sc} is the temperature value under the standard test; and η_{pv} is the output efficiency of the photovoltaic power supply.

ES is a kind of electric energy storage that plays peak-cutting and valley filling for the grid's load and ensures the system's continuous power supply. Power is electric energy storage that can charge and discharge for a short time, so the dispatching interval is not long. The electric energy storage model's relationship between the energy storage capacity and charge-discharge power can be expressed as follows:

$$P_{ES}(t) = (1 - \tau)P_{ES}(t - 1) + \left[P_{EES,ch}(t)\eta_{EES,ch} - \frac{P_{EES,dis}(t)}{\eta_{EES,dis}}\right]\Delta t \tag{4}$$

where $P_{ES}(t)$ is the energy storage capacity in period t; $P_{EES,ch}(t)$, $P_{EES,dis}(t)$, and $\eta_{EES,ch}$, $\eta_{EES,dis}$ respectively, t time to charge and discharge power and efficiency; τ is the self-discharge rate of stored energy.

MT is also called a miniature gas turbine that can be a small thermal generator with a power range of 25~300 kW, which uses natural gas, gasoline, and diesel. Its output power is controllable. Typically, the power output of a micro-gas turbine is related to the amount of fuel it uses, and the more fuel there is, the greater the power output. The mathematical model of the fuel cost of a micro gas turbine [15] is expressed as:

$$C_{MT}(t) = \frac{c_{fuel}}{LHV} \cdot \frac{P_{MT}(t)}{\eta_{MT}(t)} \tag{5}$$

where $C_{MT}(t)$ is the fuel cost of the miniature gas turbine in period t, \$; c_{fuel} is the price of natural gas, $\$/m^3$; c_{fuel} is set to 0.025 $\$/m^3$; $P_{MT}(t)$ is the output power of the micro gas turbine in period t, kW; LHV is the low calorific value of natural gas, e.g., LHV is set to 9.7 kWh/m^3; and $\eta_{MT}(t)$ is the efficiency of the micro gas turbine in period t. The relation function between the output power of MT and the power generation efficiency in the experiment section is expressed as follows:

$$\eta_{MT}(t) = 0.0753\left(\frac{P_{MT}(t)}{62}\right)^3 - 0.3095\left(\frac{P_{MT}(t)}{65}\right)^2 + 0.4174\left(\frac{P_{MT}(t)}{65}\right) + 0.1068 \tag{6}$$

FC is fuel cell power generation, which is a device that directly converts the chemical energy of the chemical reaction into electric energy with high efficiency. The power generation efficiency is much higher than other power generation methods, and it has broad prospects for application to microgrids. The fuel cost mathematical model of FC [16] is expressed as follows:

$$C_{FC}(t) = \frac{c_{fuel}}{LHV} \cdot \frac{P_{FC}(t) \cdot t}{\eta_{FC}(t)} \tag{7}$$

where $\eta_{FC}(t)$ and $P_{FC}(t)$ are respectively the efficiency and output power of the fuel cell in the period of t. The relationship between the fuel cell power generation efficiency and output power according to empirical experience [17] is expressed as follows:

$$\eta_{FC}(t) = -0.0023P_{FC}(t) + 0.6735 \tag{8}$$

An implementing microgrid needs to consider the safety grid and power balance under several constraints, consisting of the output power constraint of the. distributed power supply, the constraints of the unit generators climbing rate, and power interaction constraints. The load of the system in period t is expressed as a power balance equal constraint as follows:

$$P_{load}(t) = P_{WT}(t) + P_{PV}(t) + P_{MT}(t) + P_{FC}(t) + P_{ES}(t) + u \cdot P_{EX}(t) \tag{9}$$

where $P_{load}(t)$ is the load of the system in period t; the power load is a system supply's output power as a vector $P_i(t)$ loads of WT, $PV, \ldots,$ EX, or $i = 1.2 \ldots\ n$. The output power constraint of the distributed power supply is exppessed as follows:

$$P_{i,min} \le P_i(t) \le P_{i,max} \tag{10}$$

where $P_{i,min}$,$P_{i,max}$ are the minimum and maximum active power output of the *i-th* power supply.

The constraint of the unit climbing rate is:

$$|P_{MT}(t) - P_{MT}(t-1)| \le P_{MT}^{max} \tag{11}$$

$$|P_{FC}(t) - P_{FC}(t-1)| \le P_{FC}^{max} \tag{12}$$

where $P_{MT}(t)$, $P_{MT}(t-1)$ are the active power output of the MGT in periods t and $t-1$ respectively; $P_{FC}(t)$, $P_{FC}(t-1)$ are the active power output of the fuel cell in period t and period $t-1$, respectively; and P_{MT}^{max}, P_{FC}^{max} are the upper power limit of the fuel cell and MGT under climbing constraints, respectively.

The power interaction constraints when the microgrid and large grid are connected are:

$$P_{EX,min}(t) \leq P_{EX}(t) \leq P_{EX,max}(t) \tag{13}$$

where $P_{EX,min}(t),\ P_{EX,max}(t)$ is the minimum and maximum power exchanged between the microgrid and the large grid in the period of t.

3. Proposed ESSA Algorithm

This section presents an improved version of the sparrow search algorithm (ESSA) based on the elite reverse learning strategy and FA mutation strategy [14]. Before presenting the approach details, we will review the original algorithm of SSA [15].

3.1. Sparrow Search Algorithm

The sparrow search algorithm (SSA) is a swarm intelligence optimization algorithm based on sparrows' feeding and predator avoidance behavior [15]. It mainly simulates the process of the sparrow group foraging: the sparrow individuals who find better food act as finders, and the other individuals act as followers. Simultaneously, a certain percentage of the population is chosen to conduct reconnaissance and early warning. If danger is identified, they will give up food, and safety comes first. The position of individual sparrows is represented by the matrix below:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d} \end{bmatrix} \tag{14}$$

where n is the number of sparrows and d is the dimension of the variable to be optimized. Then, the fitness values of all sparrows can be expressed by the following vector:

$$F(X) = \begin{bmatrix} f([x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d}]) \\ f([x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d}]) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f([x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d}]) \end{bmatrix} \tag{15}$$

where *F(X)*, *n* are the fitness values of all sparrows, and the value of each row represents the fitness value of an individual and the number of sparrows. The finders are the discoverer responsible for finding food and guiding the entire population with higher fitness scores, and prioritizing obtaining food during the search. As a result, the discoverers can search for food over a much wider area than the participants. Once the sparrow detects a predator, the individual begins to sing as an alarm signal. It means when the alarm value is greater than the safety value, the finder will take the participants to other safe areas for foraging. In each iteration, the location of the sparrow finder is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot exp\left(\frac{-i}{\alpha \cdot iter_{max}}\right) & if\ R_2 < ST \\ X_{i,j}^{t} + Q \cdot L & if\ R_2 \geq ST \end{cases} \tag{16}$$

where $X_{i,j}^{t}$ is the location of the sparrow finder; t is the current iteration; $j = 1, 2, \ldots, d$ is the dimension of the *i-th* sparrow in iteration t; $iter_{max}$ is the constant with the max iterations; $\alpha \in (0, 1]$ is a random number; $R_2 (\in [0, 1])$ and $ST \in [0.5, 1]$ represent alarm values and safety thresholds, respectively; Q is a random number that follows a normal distribution; and L is set to 1 if every entry of a dimensioned matrix is 1. When $R_2 < ST$, this means there are no predators around and the finder goes into extensive search mode; otherwise, if $R_2 \geq ST$, this means that some sparrows have encountered danger with predators, and all sparrows need to quickly fly away for safety.

The lesser an entrant's energy, the worse their chances of foraging in the group as a whole. Some hungry newcomers are more inclined to flee to find more energy elsewhere. Entrants can always look for the finder during foraging, which can obtain food or forage around it. Some entrants may keep a close eye on the finders to boost their predation rate and compete for food. On the other hand, some entrants keep a closer eye on the finders if they notice that the good food discoverer will leave their current place to compete for food. If they win, they will receive the finder's food right away. The formula for the enrollees' position updates is as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot exp\left(\frac{X_{worst}^{t} - X_{i,j}^{t}}{i^2}\right) & if\ i > n/2 \\ X_P^{t+1} + \left|X_{i,j}^{t} - X_P^{t+1}\right| \cdot A^{+} \cdot L & otherwise \end{cases} \tag{17}$$

where X_{worst} is the worst position in the search space at the current moment; A^{+} is a random variable that has a dimension d with each element randomly [1, −1]; and $A^{+} = A^{T}\left(AA^{T}\right)^{-1}$. If $i > n/2$, it indicates the entrant *i-th* has a poor fitness value and is most likely to starve. About 10% to 20% are assumed to be danger aware of the sparrow population, which randomly generates the sparrows' initial positions. The sparrows at the edge of the group of the danger aware will quickly fly to the safe area to obtain a better position, while the sparrows in the middle of the group will move around randomly to get close to other sparrows. The mathematical model of the scout can be expressed as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \beta \cdot \left|X_{i,j}^{t} - X_{best}^{t}\right| & if\ f_i > f_g \\ X_{i,j}^{t} + K \cdot \left(\frac{\left|X_{i,j}^{t} - X_{worst}^{t}\right|}{(f_i - f_w) + \varepsilon}\right) & if\ f_i = f_g \end{cases} \tag{18}$$

where X_{best} is the current global optimal location; β is a step size control parameter as a normal distribution of random numbers with a mean of 0 and a variance of 1; and K is the direction of sparrow movement as the step size control coefficient that is a random number $\in [-1, 1]$. The f is the fitness function of the optimization problem, where f_i, f_g, and f_w are the current, global best, and worst sparrow fitness values, respectively; ε is the minimum constant to avoid zero division error. For simplicity, $f_i > f_g$ means sparrows are at the edge of the group, when it is safe, X_{best}, around the center; otherwise, $f_i = f_g$ indicates that sparrows in the middle of the population are aware of the danger sparrows.

3.2. Enhanced Sparrow Search Algorithm—(ESSA)

Although having several advantages, e.g., easy implementation, local search ability, and faster convergence, still, whenever dealing with a complicated problem like the grid operation scheduling cycles, the SSA algorithm also encounters the issue of a weak global search capability or jumping out of the optimal local operation or vulnerability to a local optimum. The cause means the SSA algorithm efficiency is not stable. Reverse learning and mutation are effective ways in group distribution problems to solve these limitations of the SSA algorithm [6].

3.2.1. Elite Reverse Learning Strategy

For reverse solutions of total collection from the best fitness value of sparrows by the elite strategy, a reverse group learning process [25] and an elite learning strategy group [26] have merged into a new solution set, the solution set of the fitness value of the worst sparrows, forming a new solution set. For the d-dimensional search space, let $S(x_1, x_2, \ldots, x_i \ldots, x_D)$, and $x_i \in [a_i, b_i]$, $(i = 1, 2, \ldots, d)$ is the forward solution of the problem. If a vector x is in the range $[a, b]$, then the opposite sparrows of x in the d-dimensional space can be expressed as follows:

$$x = a + b - x \tag{19}$$

The corresponding inverse vector as $S'\left(x'_1, x'_2, \ldots, x'_D\right)$, can be expressed as $x_i = a_i + b_i - x_i$. The inverse vectors of all the solutions in the optimization space are calculated, and the original forward solution set and the reverse solution set are regarded as a sort of fitness value according to the forward sparrow and the reverse sparrow as a whole. In the d-dimensioned solution space, sparrows with the best fitness value can be selected as a new optimization group through direct screening or other optimization strategies, making the sparrows in the optimization space quickly converge to the optimal solution's location. The original solution and reverse solution vectors of collection using the elite strategy generate new solutions with a specific rate to join the original solution and inverse solution set, thus obtaining the new optimization group. A new solution X_{inew} is produced for the optimized mathematical form as follows:

$$X_{inew} = X_i \times Q_1 \tag{20}$$

$$Q_1 = R_{istar} \times \frac{rand(-0.5, 0.5)}{D} \tag{21}$$

where Q_1 is the change factor for generating new solutions; D is the dimension of the solution space; R_{istar} is the Euclidean distance between the optimal solution and the nearest solution to the optimal solution; and $rand(-0.5,\ 0.5)$ is a random number between -0.5 and 0.5. After sorting the fitness values of the solution vectors in the new set, the 20% and d-dimension solutions X_{iWorst} with the worst fitness values are eliminated to generate a new optimization group.

3.2.2. Firefly Algorithm Mutation Strategy

In the firefly algorithm (FA) [14], individual fireflies emit light, which acts as a signal to attract other individual fireflies. The FA shows several advantages of optimization processing that can be used for enhancing SSA's performance, e.g., search ability with its mutations, fast convergence, and fewer parameters easy to operate. A mutation strategy equation is one of the FA's characteristics used to hybridize SSA's updating formula in Equation (18), which generates new solutions as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta_1 \cdot \left| X_{i,j}^t - X_{best}^t \right| + \alpha \cdot \left(rand - \frac{1}{2} \right) & if\ f_i > f_g \\ X_{i,j}^t + K \cdot \left(\frac{\left| X_{i,j}^t - X_{worst}^t \right|}{(f_i - f_w) + \varepsilon} \right) & if\ f_i = f_g \end{cases} \tag{22}$$

where X_{best} is the global optimal position of the current sparrow; β_1 is the step size control parameter; $K \in [-1,\ 1]$ is a random number; f_i f_g, and f_w are the current fitness value, current global optimal fitness value, and current global worst fitness value; ε is a minimum constant to avoid zero division error; $\alpha \in [0,\ 1]$ is a step size factor; and $rand \in [0,\ 1]$ is uniformly distributed random numbers.

3.3. ESSA Algorithm Evaluations

To verify the feasibility and potential of the proposed ESSA algorithm, we selected several specific test functions of the CEC2019 test suit [27]. The proposed ESSA algorithm is fully investigated through various benchmark functions that include multi-modals with high single-peak, high multi-peak, and low-dimensional multi-peak to test its performance. The selected test function set parameters are listed in Table 1.

Table 1. Selected benchmark functions.

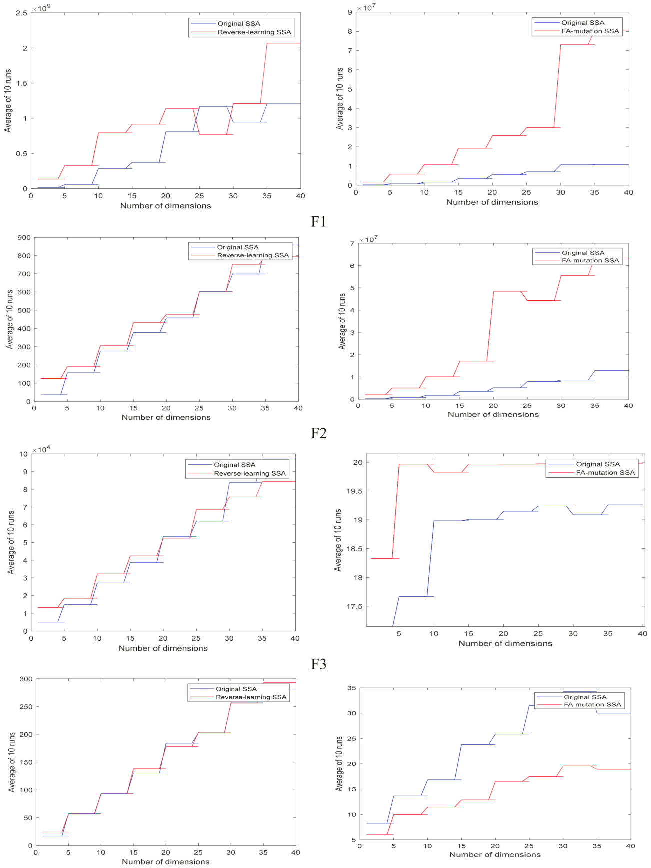
No.	Function Name	Function	Dim	Space	f_{min}
F1	Sphere	$\sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0
F2	Schwefel's function 2.21	$\sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	30	$[-10, 10]$	0
F3	Schwefel's function 1.2	$\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	$[-100, 100]$	0
F4	Schwefel's function 2.22	$max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	$[-100, 100]$	0
F5	Dejong's noisy	$\sum_{i=1}^{n} ix_i^4 + random[0, 1)$	30	$[-100, 100]$	0
F6	Schwefel	$\sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	$[-500, 500]$	$-125{,}969$
F7	Rastringin	$\sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	$[-5.12, 5.12]$	0
F8	Ackley	$-20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 + e$	30	$[-32, 32]$	0
F9	Griewank	$\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x[i]}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
F10	Generalized penalized 2	$\left(\frac{1}{500} + \sum_{j=1}^{25}\left(j + \sum_{i=1}^{2}(x_i - a_{ij})^6\right)^{-1}\right)^{-1}$	30	$[-50, 50]$	0
F11	Rosenbrock	$\sum_{i=1}^{n-1}\left[100\left(x_{i+1} - x_i^2\right)^2 - (x_i - 1)^2\right]$	30	$[-30, 30]$	0
F12	Sphere- steps	$\sum_{i=1}^{n}(x_i + 0.5)^2$	30	$[-100, 100]$	0

We compared each of the two mechanisms of the elite reverse-learning (strategy 1) and FA mutation (strategy 2) with the original SSA and the ESSA (both strategies 1 and 2) in the average outcomes and executed time to verify their effect on the proposed ESSA. Table 2 depicts the comparison of average outcomes and executed time of each of the two mechanisms of elite reverse-learning (strategy 1) and FA-mutation (strategy 2) with the original SSA and the ESSA. For the high single-peak benchmark functions, the execution time of the ESSA is longer than the other strategy-applied algorithms. Still, its execution time is the same as the different methods for the multi-modals and dimension multi-peak.

Table 2. Comparison of average outcomes and executed time of each of the two mechanisms of the elite reverse-learning (strategy 1) and FA-mutation (strategy 2) with the original SSA and the ESSA (strategies 1 and 2).

Algorithms	Strategy 1 Reverse-Learning SSA		Strategy 2 FA-Mutation SSA		Original (SSA)		Strategies 1&2 (ESSA)	
	Average	Exe.Time	Average	Exe.Time	Average	Exe.Time	Average	Exe.Time
F1	3.0×10^{0}	23.0	2.8×10^{-34}	21.9	2.6×10^{-41}	23.9	2.6×10^{-67}	24.4
F2	1.0×10^{0}	13.2	2.3×10^{-13}	12.4	9.7×10^{-41}	13.5	9.6×10^{-41}	13.8
F3	2.2×10^{1}	12.6	8.2×10^{-14}	11.4	2.3×10^{-34}	12.5	2.3×10^{-56}	12.7
F4	4.4×10^{-1}	13.0	8.5×10^{-16}	12.4	1.1×10^{-8}	13.5	1.1×10^{-38}	13.8
F5	1.9×10^{0}	44.0	1.8×10^{-3}	41.8	1.6×10^{-3}	45.7	1.6×10^{-3}	46.6
F6	-7.0×10^{1}	123.0	-1.2×10^{1}	116.9	-1.1×10^{3}	117.8	-1.2×10^{1}	130.4
F7	9.6×10^{1}	23.0	3.3×10^{2}	21.9	2.2×10^{-1}	23.9	2.2×10^{-1}	24.4
F8	3.7×10^{0}	32.2	3.3×10^{-15}	30.4	2.1×10^{-16}	33.2	2.1×10^{-16}	33.9
F9	4.2×10^{1}	12.0	5.8×10^{0}	11.4	8.7×10^{-1}	12.5	8.7×10^{-1}	12.7
F10	4.4×10^{-1}	12.0	6.2×10^{0}	11.4	2.9×10^{-1}	12.5	2.9×10^{-1}	12.7
F11	-3.6×10^{0}	32.0	-5.9×10^{0}	30.4	-2.4×10^{0}	33.2	-2.4×10^{0}	33.9
F12	-1.4×10^{1}	43.0	-1.2×10^{1}	40.9	-2.3×10^{1}	44.7	-2.3×10^{1}	41.6

Figure 2 shows the effective applied equation strategies of each of the two mechanisms of the elite reverse-learning (strategy 1) and FA-mutation (strategy 2) with the original SSA for the selected test functions. Strategy 1 of the elite reverse-learning could enhance the algorithm's exploiting ability as the local search. In contrast, strategy 2 of the FA-mutation makes more diverse solutions and increases the algorithm's exploring ability.



F1

F2

F3

F4

Figure 2. The effective applied equation strategies of each of the two mechanisms of the elite reverse-learning (strategy 1) and FA-mutation (strategy 2) with the original SSA in different problem dimension spaces.

The obtained results of the proposed algorithm are compared with the other methods in the literature, e.g., PSO [17], GA [16], FA [14], BA [20], GWO [21], and SSA [15] algorithms. The setting parameters for the algorithms in the experiment with the same condition and platform environment, e.g., population size N set to 30, maximum iteration times T is set to 500, dimension D of the test function, and upper and lower bounds u_b and l_b of the initial value are set according to the reference functions in Table 1. The PSO's weight parameter is set to 0.4 to 0.9; the factors of c_1 and c_2 are set to 1.7 [17]. The FA's randomization parameter α ranges in [0 to 1], attractiveness β_0 is set to 0.02, and the absorption coefficient γ is set to 0.7 [14]. The BA's frequency f_{min}, f_{max} is set to the range [0, 5], A_0 is set to 0.92, and α and γ are set to 0.9 and 0.98, respectively [20]. The GWO's coefficient of the prey and search wolf position vectors are initialized in the range [0, 1]; the variables of the arbitrary values r_1 and r_2 are set to drop from 2 to 0 over the iteration courses. The number of finders $pNum$ and the number of the reconnaissance and warning sparrows $sNum$ are both 20% of the population size [15].

The evaluation indexes of the comparable outcomes are the obtained mean and standard deviation values from the algorithms for the selected benchmark functions. The experimental results are an average of the obtained results of 30 independent runs for each benchmark function to avoid the contingency of the optimization results and prove the stability of the suggested algorithm, so the number of runs is set to 30.

Tables 3–5 show the experimental data of the obtained results of the proposed algorithm compared with the other methods, e.g., PSO, GA, FA, BA, GWO, and SSA algorithms. It can be seen that the proposed algorithm produces the optimization results of functions, e.g., F1–F5, F8, and F10–F11, better than the other algorithms in terms of the optimization accuracy, and the optimization results of F1–F4 and F10–F12 of ESSA are significantly improved compared with the original algorithm.

Table 3. Obtained optimization results comparison of the proposed ESSA with the PSO and GA for the benchmark functions.

Algorithms	PSO		GA		ESSA	
	Average	Sd.	Average	Sd.	Average	Sd.
F1	5.07×10^{0}	1.72×10^{0}	1.63×10^{-2}	9.80×10^{-3}	6.20×10^{-68}	3.19×10^{-77}
F2	6.92×10^{0}	2.73×10^{0}	2.52×10^{-2}	9.80×10^{-3}	1.77×10^{-40}	4.16×10^{-40}
F3	1.43×10^{2}	6.54×10^{2}	2.65×10^{2}	2.40×10^{2}	3.31×10^{-65}	1.31×10^{-64}
F4	5.16×10^{0}	1.41×10^{0}	1.50×10^{0}	6.45×10^{-1}	5.19×10^{-29}	2.24×10^{-38}
F5	1.27×10^{1}	9.09×10^{0}	1.94×10^{-2}	8.23×10^{-3}	7.24×10^{-4}	6.41×10^{-4}
F6	-3.21×10^{3}	4.49×10^{2}	-5.46×10^{3}	9.53×10^{2}	-1.11×10^{4}	7.13×10^{2}
F7	1.90×10^{2}	4.05×10^{1}	4.30×10^{1}	1.80×10^{1}	0.00×10^{0}	0.00×10^{0}
F8	3.04×10^{0}	3.85×10^{-1}	2.56×10^{-2}	9.39×10^{-3}	8.88×10^{-16}	0.00×10^{-0}
F9	2.98×10^{1}	8.58×10^{0}	2.50×10^{-1}	1.23×10^{-1}	0.00×10^{0}	0.00×10^{0}
F10	3.57×10^{0}	2.22×10^{0}	5.60×10^{0}	4.13×10^{0}	1.16×10^{0}	5.27×10^{-1}
F11	-3.27×10^{0}	5.92×10^{-2}	-3.22×10^{0}	8.40×10^{-2}	-3.30×10^{0}	4.12×10^{-2}
F12	-8.55×10^{0}	3.38×10^{0}	-9.75×10^{0}	2.23×10^{0}	-1.02×10^{1}	1.29×10^{-5}

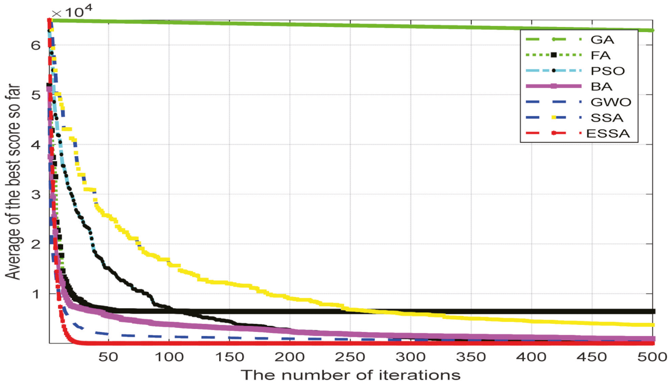
Table 4. Obtained optimization results comparison of the proposed ESSA with the FA and SSA for the benchmark functions.

Algorithms	FA		SSA		ESSA	
	Average	Sd.	Average	Sd.	Average	Sd.
F1	3.07×10^{0}	1.72×10^{0}	3.76×10^{-24}	2.06×10^{-23}	6.40×10^{-78}	3.19×10^{-77}
F2	3.92×10^{0}	2.73×10^{0}	1.67×10^{-13}	7.28×10^{-13}	1.87×10^{-40}	4.16×10^{-40}
F3	1.33×10^{1}	6.54×10^{1}	6.53×10^{-14}	3.31×10^{-13}	3.21×10^{-65}	1.31×10^{-64}
F4	5.16×10^{-1}	1.41×10^{-1}	6.98×10^{-16}	3.28×10^{-15}	5.29×10^{-49}	2.24×10^{-28}
F5	1.27×10^{1}	9.09×10^{0}	4.25×10^{-3}	4.38×10^{-3}	7.24×10^{-4}	6.41×10^{-4}
F6	-3.21×10^{3}	4.49×10^{2}	-8.51×10^{3}	6.87×10^{2}	-1.114×10^{1}	7.13×10^{1}
F7	1.90×10^{2}	4.05×10^{1}	2.27×10^{2}	3.87×10^{1}	0.00×10^{0}	0.00×10^{0}
F8	3.04×10^{0}	3.85×10^{-1}	1.48×10^{-15}	1.89×10^{-15}	9.88×10^{-16}	0.00×10^{0}
F9	1.98×10^{1}	1.58×10^{0}	4.74×10^{1}	5.37×10^{1}	0.00×10^{0}	0.00×10^{0}
F10	2.57×10^{0}	3.22×10^{0}	5.55×10^{0}	5.22×10^{0}	1.16×10^{0}	5.27×10^{-1}
F11	-2.27×10^{0}	5.92×10^{-2}	-3.27×10^{0}	6.03×10^{-2}	-3.30×10^{0}	4.12×10^{-2}
F12	-7.55×10^{0}	3.38×10^{0}	-7.65×10^{0}	2.74×10^{0}	-1.02×10^{1}	1.29×10^{-5}

Table 5. Obtained optimization results comparison of the proposed ESSA with the BA and GWO for the benchmark functions.

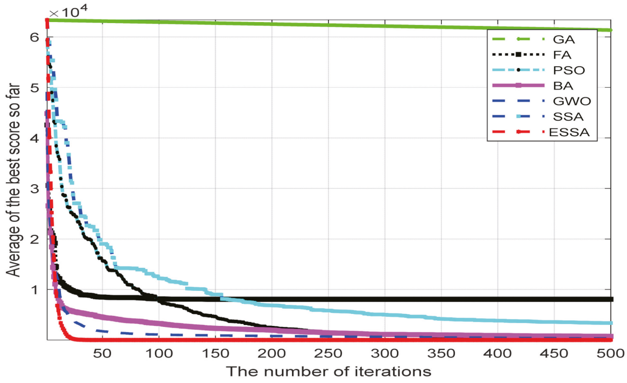
Algorithms	BA		GWO		ESSA	
	Average	Sd.	Average	Sd.	Average	Sd.
F1	1.89×10^{0}	1.70×10^{0}	1.75×10^{-24}	1.09×10^{-23}	1.64×10^{-78}	4.77×10^{-78}
F2	6.26×10^{-1}	2.26×10^{0}	1.45×10^{-13}	1.02×10^{-12}	6.02×10^{-41}	3.20×10^{-41}
F3	1.35×10^{1}	2.34×10^{1}	5.12×10^{-14}	2.61×10^{-14}	1.41×10^{-65}	1.90×10^{-64}
F4	2.77×10^{-1}	1.23×10^{-1}	5.31×10^{-16}	2.51×10^{-16}	7.11×10^{-39}	1.97×10^{-39}
F5	1.18×10^{0}	6.83×10^{0}	1.10×10^{-3}	5.61×10^{-3}	9.86×10^{-4}	3.39×10^{-6}
F6	-4.48×10^{3}	5.41×10^{2}	-7.18×10^{3}	4.45×10^{2}	-6.99×10^{1}	4.59×10^{2}
F7	5.99×10^{1}	1.12×10^{1}	2.06×10^{2}	1.67×10^{1}	1.37×10^{-1}	2.28×10^{-3}
F8	2.30×10^{0}	4.70×10^{-1}	2.08×10^{-15}	1.08×10^{-15}	1.33×10^{-16}	4.14×10^{-3}
F9	2.60×10^{1}	2.19×10^{0}	3.65×10^{0}	2.00×10^{1}	0.01×10^{0}	6.96×10^{-3}
F10	2.78×10^{-1}	3.86×10^{0}	3.86×10^{0}	5.18×10^{0}	1.81×10^{-1}	7.43×10^{-2}
F11	-2.26×10^{0}	5.49×10^{-2}	-3.71×10^{0}	8.19×10^{-2}	-1.49×10^{0}	1.75×10^{-2}
F12	-8.71×10^{0}	3.33×10^{0}	-7.50×10^{0}	8.58×10^{-1}	-1.45×10^{1}	1.13×10^{-5}

To reflect the dynamic convergence characteristics of ESSA, the convergence curves of the algorithms for the selected benchmark functions are obtained. Figure 3 shows the comparison of the proposed ESSA convergence curves with the other algorithms, e.g., SSA, BA, GWO, FA, PSO, and GA, obtained on the selected benchmark functions. It can be seen from the convergence curve that ESSA is significantly better than the other algorithms in terms of the convergence speed and optimization precision. The compared results indicate that ESSA could be a potential method with a searching ability while ensuring the exploration ability without losing the diversity of the population and optimization stability.

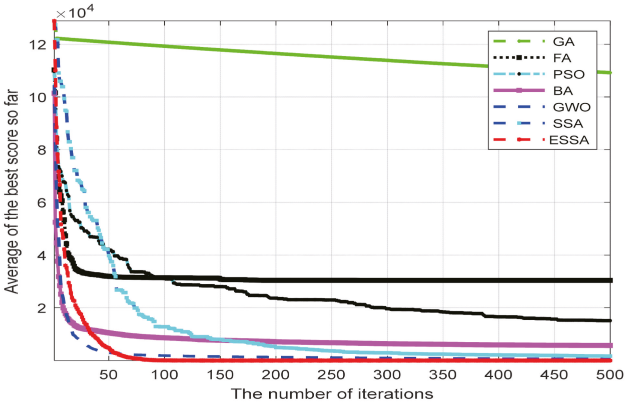


F1

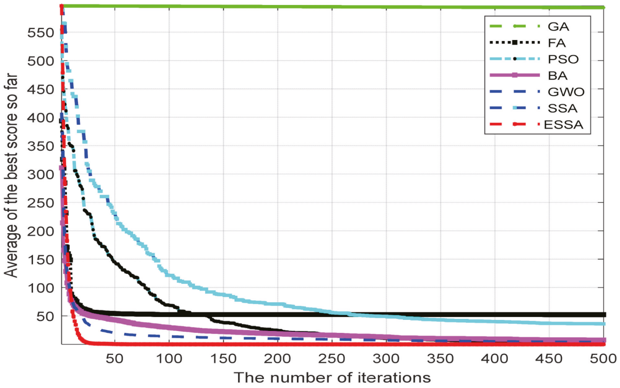
Figure 3. *Cont.*



F2

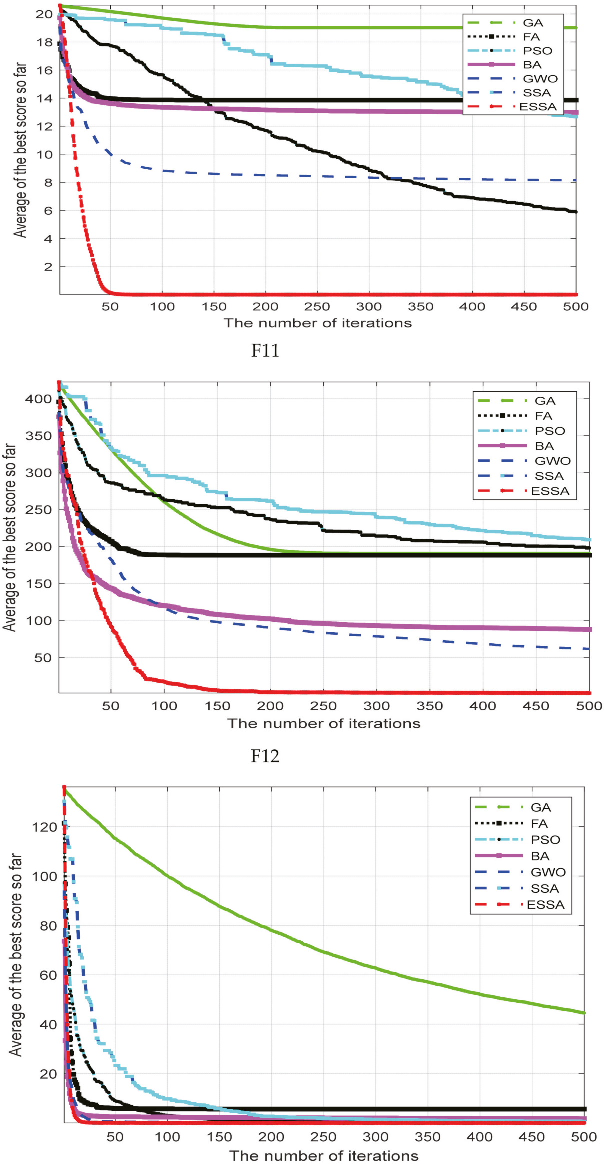


F3



F4

Figure 3. *Cont.*



F11

F12

Figure 3. Comparison of convergence curves of seven algorithms obtained on the selected benchmark functions, e.g., F1–F4 and F11–F12.

4. Applied ESSA for Power Microgrid Operations Planning

The microgrid's mathematical model of optimal operation is established based on the total operation cost in a microgrid scheduling cycle [23,24]. The objective function by the mathematical modeling of a power microgrid is implemented to find out the feasible optimization area in the problem search space by applying the proposed ESSA. A flowchart of the ESSA for planning microgrid operations is shown in Figure 4.

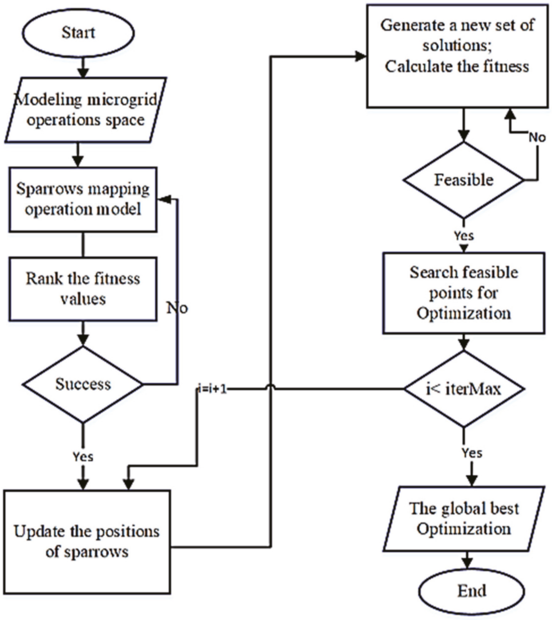

Start
Modeling microgrid operations space
Sparrows mapping operation model
Rank the fitness values
Success
No
Yes
Update the positions of sparrows
i=i+1
Generate a new set of solutions; Calculate the fitness
No
Feasible
Yes
Search feasible points for Optimization
i< iterMax
Yes
The global best Optimization
End

Figure 4. Flowchart of the ESSA for planning microgrid operations.

4.1. The Objective Function

The objective function is modeled based on the lowest power generation cost (considering the sum-up of fuel cost, depreciation cost, maintenance cost, energy interaction cost, and environmental cost) in the grid optimization cycle. The output of the micropower supply is calculated according to the unit of the microgrid of the daily load curve and the wind-scene output curve with the time interval with the period. The operation parameters relate to each treatment cost and emission coefficient of various pollutants, and the time-of-use price. The day of the time interval and year times is divided into 24 h periods (each optimization period is an hour) and 12 months.

The data values or curves of the wind speed and light solar intensity are found from the predicted weather forecast of a certain day and place. The active power's electricity price with the large power grid and the variable electricity price change synchronously within each optimization period. The mathematical modeling of a power microgrid in establishing microgrid optimization operation for the objective function can be expressed as follows:

$$min\ F = \omega \times C_1 + (1-\omega) \times C_2 \tag{23}$$

where F is the objective function; C_1 and C_2 are the power generation and environment deployed costs, respectively, in the optimization model; and ω is the weight variable (ω is set to 0.5 in the experiment). The power generation cost of a microgrid is distributed loads as follows:

$$C_1 = \sum_{t=1}^{T}[C_{FC}(t) + C_{DP}(t) + C_{ME}(t) + \mu \cdot C_{EX}(t)] \tag{24}$$

where μ = 1 for grid-connected operation; μ = 0 for off-grid operation; and $C_{FC}(t)$, $C_{DP}(t)$, $C_{ME}(t)$, $C_{EX}(t)$ are the fuel cost, investment depreciation cost, maintenance cost, and interaction power cost with the large power grid in the period of t, respectively. A microgrid development environment's conversion cost is calculated as follows:

$$C_2 = \sum_{t=1}^{T} \sum_{k=1}^{K} b_k \left(\sum_{i=1}^{N} a_{i,k} \cdot P_i \right) \tag{25}$$

where C_2 is the cost of treating pollutants discharged from a microgrid; K is the serial number of pollutants discharged by each distributed power source; b_k is the cost of treatment of class K pollutants, \$/kg; and $a_{i,k}$ is the coefficient of class K pollutants discharged by the ith distributed power source, *g/KWh*.

4.2. Microgrid Operations Planning

Figure 4 shows the flowchart of the ESSA for planning microgrid operations. It means that the proposed approach is applied to solve microgrid scheduling cycles with the power source's optimal output and total operation cost.

The main steps of the ESSA algorithm for planning microgrid operations are listed as follows:

- Step 1. Input system model parameters of a microgrid operation, daily load and microgrid output curves, unit generating set, time-of-use electricity price, and various pollution cost treatment coefficients.
- Step 2. Initialize population sparrows randomly, and calculate the fitness value of each sparrow by using the objective function. A new solution set is formed by selecting sparrows with the best fitness value from the total set of forward and reverse solutions and combining them into the solution set according to the elite strategy. Selected sparrows with the worst fitness value in the solution set are removed to form a new set of solutions.
- Step 3. Rank the fitness to find the current best fitness individual and the worst fitness individual.
- Step 4. Update the positions of sparrows with higher fitness and sparrows with lower fitness, and randomly update the positions of some sparrows to get the current updated positions.
- Step 5. Check the better sparrow positions: if the new position is superior to the old position, update the old position.
- Step 6. Calculate the fitness value of the sparrow positions and then generate a new set of solutions by the reverse elite learning strategy and preserve the global and historical optimal values.
- Step 7. Check the termination condition, e.g., if it reaches max-iteration, repeat steps 2 to 6; otherwise, output the best outcome value and best sparrow positions.

4.3. Analysis and Discussion Results

The setting parameters for the algorithms in the experiment with the same condition and platform environment are set, e.g., population size N set to 30, maximum iteration times is set to 1000, a number of runs are set to 25. Table 6 lists the operating parameters of each unit of the microgrid system. The operational parameters are used as the inputs with the boundary search space in the microgrid issue. Figure 5 displays a scenario of the demand daily local load curves of the entire power system on the island.

Table 6. Operating parameters of each unit of the microgrid system.

Micro Power Types	Power Capacity/kW		Climb Rate Constraint	Equipment Maintenance Factor	Installation Costs (10^3 $/KW)	The Capacity Factor/%
	Upper	Lower				
WT	40.4	0	0.001	0.0296	2.37	22.13
PV	30.5	0	0.001	0.0096	6.65	29.34
MT	60.1	15	10	0.088	1.306	55.94
FC	40.2	5	2	0.087	4.275	30.34
ES	50.3	−50	0.0001	0.004	0.087	32.67
MG	60.4	−60	0.001	0.001	0.0001	0.002

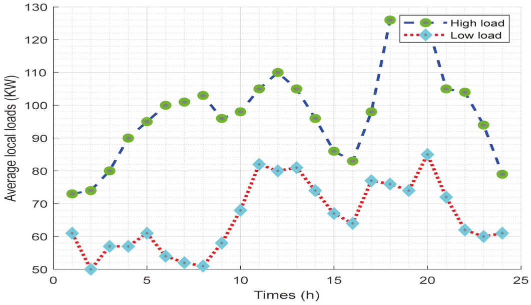


Figure 5. Daily demand local load curves of the power system in the island.

The other parameters, e.g., the treatment cost, emission coefficient of various pollutants, and price of the microgrid, are shown in Tables 6 and 7, respectively.

Table 7. Pollutant treatment costs and emission coefficients.

Emissions		CO_2	SO_2	NO_x	CO
Discharge coefficient/$(g{\cdot}kW^{-1})$	MT	184	0.00093	0.619	0.17
	FC	635	0	0.023	0.054
Processing cos ts/$(\$/kg^{-1})$		0.0041	0.875	1.25	0.145

The cost of purchasing electricity from the large grid in the grid-connected state is lower than the cost of generating electricity from the microgrid in the trough, so electricity is purchased from the large grid. The consumption power to the large grid during peak load times and during peak load times to meet the load demand generates revenue for the microgrid. Table 8 lists the microgrid TOU price meter.

Table 8. Microgrid TOU price meter.

Periods	Period of Time	Price/($/KW·h)
Normal period	07:00–10:00	0.49
	15:00–18:00	
	21:00–23:00	
Peak period	10:00–15:00	0.83
	18:00–21:00	
Trough period	23:00–07:00	0.17

In the microgrid, the battery is charged at the trough to satisfy the peak load demand. During peak loads, miniature gas turbines and fuel cells work at maximum power. The micro-grid is powered by battery discharge when it is off the grid. When the system generates excess electric energy, the battery is charged to ensure the system's continuous power supply.

The operation scenarios include grid-connected and off-grid operations impacted by year seasons, such as windy and solar light, rainy and dry seasons, or capacity facilities, such as load shift on long-term capacity planning based on historical data and load demand curves. Seasonal and daily complimentary features of energy resources express the power-generating output as a per-unit value. The benchmark capacity, on the other hand, is expressed in terms of the installed capacity. Wind and solar PV power create yearly and seasonal outputs that are highly dependent on wind and solar resources. The outcomes of the proposed scheme of the ESSA are compared with the other schemes in the literature, e.g., the PSO [28], FA [29], and SSA algorithms.

Table 9 shows the comparison of the outcomes of the proposed scheme of the ESSA with the other schemes in the literature, e.g., the PSO, FA, and SSA algorithms. It can be seen that the grid connection and off-grid optimization results from the proposed scheme products of the figure outperform the other schemes. Figures 6 and 7 show the typical daily and monthly recourse load outputs of a microgrid system.

Table 9. Grid-connected and off-grid operation optimization results.

Operation Types	A Grid-Connected Operation				A Off-Grid Operation			
Algorithms	**FA**	**PSO**	**SSA**	**ESSA**	**FA**	**PSO**	**SSA**	**ESSA**
Optimization results	810.25	820.15	788.46	718.93	842.19	852.19	969.88	792.51
Number of convergence	310	320	299	233	282	262	375	242

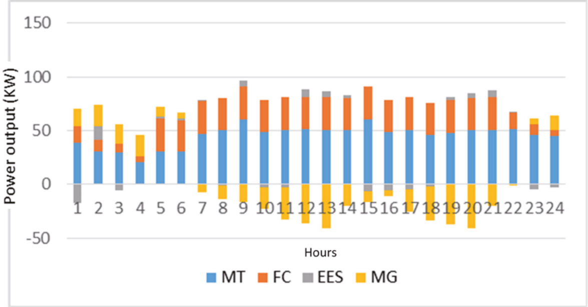


Figure 6. Typical daily recourse load outputs of a microgrid system.

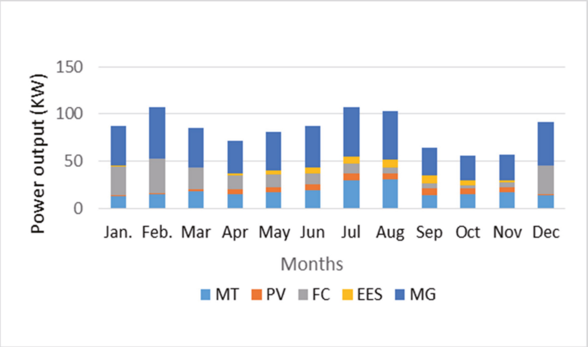


Figure 7. Typical monthly recourse load outputs of a microgrid system.

Figures 8 and 9 compare the optimization results obtained by the proposed ESSA with the FA, PSO, and SSA schemes for planning schedule cycles for daily and monthly scales, respectively. The curve of the obtained optimization results from the proposed ESSA has a faster archived convergence speed than the other schemes, e.g., FA, PSO, and SSA, in the scenario of the objective function in optimizing the grid operation planning.

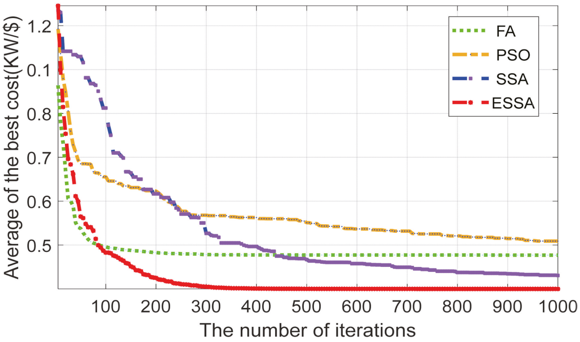


Figure 8. Comparison of the optimization results obtained by the proposed ESSA with the FA, PSO, and SSA schemes in daily schedule cycles.

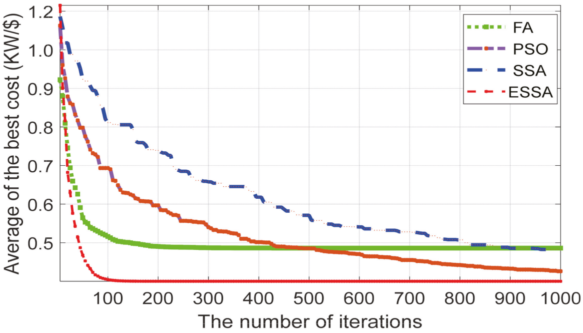


Figure 9. Comparison of the optimization results obtained by the proposed ESSA with the FA, PSO, and SSA schemes in monthly schedule cycles.

Figures 10 and 11 show the daily load and the microgrid component-distributed power sources' output curves of grid-connected and off-grid optimization.

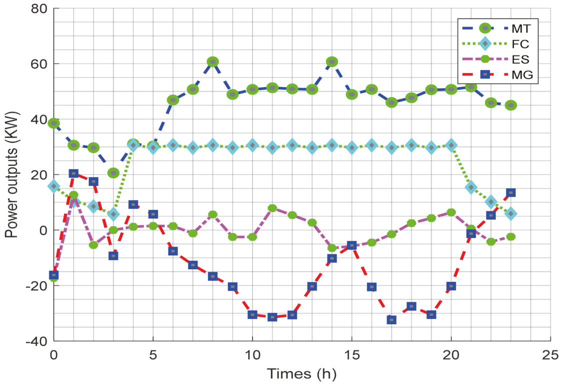


Figure 10. The daily load and the microgrid component-distributed power sources' output curves of grid-connected and off-grid optimizations.

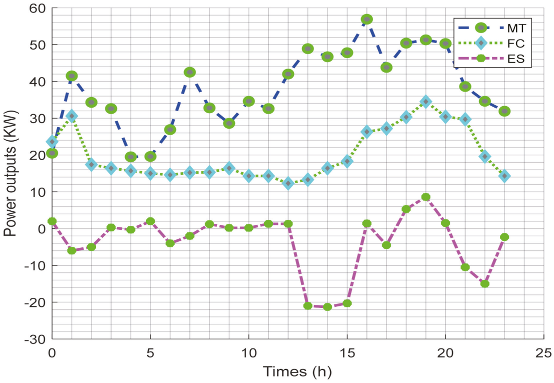


Figure 11. The daily load and the microgrid component-distributed power sources' output curves of off-grid optimizations.

In general, the observed tables and figures of the compared results show that the proposed plan reduces the net loss of the power system and the consumption of fuel generation sets. Conserved energy sources meet the power demand, safety system protection constraints, and stable power microgrid system.

5. Conclusions

This paper proposed an enhanced sparrow search algorithm (ESSA) using elite reverse learning and Firefly algorithm (FA) mutation strategies for optimal power microgrid operations planning. The operations planning in the microgrid of the power system is significant not only in dispatch economic but also in balancing distribution symmetry power sources, including wind turbines, photovoltaic, energy storage systems, microturbines, fuel cells, and various parts of the load set. We verified the proposed ESSA performance of enhancing search accuracy and convergence speed by testing the selected benchmark functions and compared the testing results with the other methods in the literature. The mathematical model of optimal operations planning of the microgrid schedule was established based on the grid-connected and off-grid microgrids for a distributed power source's optimal output and total operation cost in a microgrid scheduling cycle. The mathematical operating model considered as the objective function was calculated for the optimization by applying the ESSA. The compared results show the suggested method is feasibile and effective with high precision and a fast convergence speed regarding its optimization ability and outstanding application prospects.

Author Contributions: Conceptualization, T.-T.N. and T.-K.D.; methodology, T.-T.N.; software, T.-K.D.; validation, T.-T.-T.N., T.-K.D. and T.-G.N.; formal analysis, T.-K.D.; investigation, T.-T.N.; resources, T.-K.D.; writing—original draft preparation, T.-G.N.; writing—review and editing, T.-T.N.; visualization, T.-G.N.; supervision, T.-T.-T.N.; project administration, T.-G.N.; funding acquisition, T.-K.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zeng, Z.; Zhao, R.; Yang, H.; Tang, S. Policies and demonstrations of micro-grids in China: A review. *Renew. Sustain. Energy Rev.* **2014**, *29*, 701–718. [CrossRef]
2. Martin-Martínez, F.; Sánchez-Miralles, A.; Rivier, M. A literature review of Microgrids: A functional layer based classification. *Renew. Sustain. Energy Rev.* **2016**, *62*, 1133–1153. [CrossRef]
3. Hongtao, L.; Wenjia, L. The analysis of effects of clean energy power generation. *Energy Procedia* **2018**, *152*, 947–952. [CrossRef]
4. Kaur, A.; Kaushal, J.; Basak, P. A review on microgrid central controller. *Renew. Sustain. Energy Rev.* **2016**, *55*, 338–345. [CrossRef]
5. Ngo, T.-G.; Nguyen, T.-T.T.; Nguyen, T.-X.H.; Nguyen, T.-D.; Do, V.-C.; Nguyen, T.-T. A Solution to Power Load Distribution Based on Enhancing Swarm Optimization BT—Advances in Engineering Research and Application. In Proceedings of the International Conference on Engineering Research and Applications (ICERA), Thai Nguyen, Vietnam, 1–2 December 2020; Sattler, K.-U., Nguyen, D.C., Vu, N.P., Long, B.T., Puta, H., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 53–63.
6. Nguyen, T.-T.; Wang, H.-J.; Dao, T.-K.; Pan, J.-S.; Liu, J.-H.; Weng, S.-W. An Improved Slime Mold Algorithm and Its Application for Optimal Operation of Cascade Hydropower Stations. *IEEE Access* **2020**, *8*, 226754–226772. [CrossRef]
7. Tsai, C.F.; Dao, T.K.; Pan, T.S.; Nguyen, T.T.; Chang, J.F. Parallel bat algorithm applied to the economic load dispatch problem. *J. Internet Technol.* **2016**, *17*, 761–769. [CrossRef]
8. Dao, T.K.; Pan, T.S.; Nguyen, T.T.; Chu, S.C. Evolved Bat Algorithm for solving the Economic Load Dispatch problem. In Proceedings of the Advances in Intelligent Systems and Computing, Nanchang, China, 18–20 October 2014; Volume 329, pp. 109–119.
9. Baziar, A.; Kavousi-Fard, A. Considering uncertainty in the optimal energy management of renewable micro-grids including storage devices. *Renew. Energy* **2013**, *59*, 158–166. [CrossRef]
10. Cabrera-Tobar, A.; Bullich-Massagué, E.; Aragüés-Peñalba, M.; Gomis-Bellmunt, O. Review of advanced grid requirements for the integration of large scale photovoltaic power plants in the transmission system. *Renew. Sustain. Energy Rev.* **2016**, *62*, 971–987. [CrossRef]
11. Nguyen, T.T.; Wang, M.J.; Pan, J.S.; Dao, T.K.; Ngo, T.G. A Load Economic Dispatch Based on Ion Motion Optimization Algorithm. In Proceedings of the Smart Innovation, Systems and Technologies, Jilin, China, 18–20 July 2019; Volume 157, pp. 115–125.
12. Pan, J.; Liu, N.; Chu, S. A Hybrid Differential Evolution Algorithm and Its Application in Unmanned Combat Aerial Vehicle Path Planning. *IEEE Access* **2020**, *8*, 17691–17712. [CrossRef]
13. Abdel-Basset, M.; Abdel-Fatah, L.; Sangaiah, A.K. Metaheuristic algorithms: A comprehensive review. In *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*; Universitat Politècnica de Catalunya: Barcelona, Spain, 2018; pp. 185–231. ISBN 9780128133149.
14. Yang, X.-S. Firefly Algorithm, Levy Flights and Global Optimization. In *Research and Development in Intelligent Systems XXVI*; Springer: London, UK, 2010; pp. 209–218. [CrossRef]
15. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
16. Tomassini, M. A Survey of Genetic Algorithms. *Annu. Rev. Comput. Phys. World Sci.* **1995**, *3*, 87–118.
17. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
18. Yang, X.-S. Harmony search as a metaheuristic algorithm. In *Music-Inspired Harmony Search Algorithm*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–14.
19. Dorigo, M.; Di Caro, G. Ant colony optimization: A new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; Volume 2, pp. 1470–1477.

20. Yang, X.-S.; Hossein Gandomi, A. Bat algorithm: A novel approach for global engineering optimization. *Eng. Comput. Int. J. Comput. Eng. Softw.* **2012**, *29*, 464–483. [CrossRef]
21. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
22. Hruschka, E.R.; Campello, R.J.G.B.; Freitas, A.A. A survey of evolutionary algorithms for clustering. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2009**, *39*, 133–155. [CrossRef]
23. Fu, Q.; Nasiri, A.; Solanki, A.; Bani-Ahmed, A.; Weber, L.; Bhavaraju, V. Microgrids: Architectures, controls, protection, and demonstration. *Electr. Power Compon. Syst.* **2015**, *43*, 1453–1465. [CrossRef]
24. Huang, W.-T.; Tai, N.-L.; Fan, C.-J.; Lan, S.-L.; Tang, Y.-Z.; Zhong, Y. Study on structure characteristics and designing of microgrid. *Power Syst. Prot. Control* **2012**, *40*, 149–155.
25. Hassard, A. Reverse learning and the physiological basis of eye movement desensitization. *Med. Hypotheses* **1996**, *47*, 277–282. [CrossRef]
26. Xia, X.; Tang, Y.; Wei, B.; Gui, L. Dynamic multi-swarm particle swarm optimization based on elite learning. *IEEE Access* **2019**, *7*, 184849–184865. [CrossRef]
27. Liang, J.J.; Qu, B.Y.; Gong, D.W.; Yue, C.T. *Problem Definitions and Evaluation Criteria for the CEC 2019 Special Session on Multimodal Multiobjective Optimization*; The Computational Intelligence Laboratory, Zhengzhou University: ZhengZhou, China, 2019.
28. Ma, Y.; Yang, P.; Zhao, Z.; Wang, Y. Optimal economic operation of islanded microgrid by using a modified pso algorithm. *Math. Probl. Eng.* **2015**, *2015*, 379250. [CrossRef]
29. Amamra, S.-A.; Ahmed, H.; El-Sehiemy, R.A. Firefly algorithm optimized robust protection scheme for DC microgrid. *Electr. Power Compon. Syst.* **2017**, *45*, 1141–1151. [CrossRef]

symmetry

MDPI

Article

Design and Optimization of Combined Cooling, Heating, and Power Microgrid with Energy Storage Station Service

Nan Ning [1,2], Yu-Wei Liu [1,2,*], Hai-Yue Yang [3] and Ling-Ling Li [1,2]

[1] State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300130, China; 202031404042@stu.hebut.edu.cn (N.N.); lilingling@hebut.edu.cn (L.-L.L.)
[2] Key Laboratory of Electromagnetic Field and Electrical Apparatus Reliability of Hebei Province, Hebei University of Technology, Tianjin 300130, China
[3] Hengshui Power Supply Branch of State Grid Hebei Power Co., Ltd., Hengshui 053000, China; hengshuiyhy@gmail.com
* Correspondence: 201811401014@stu.hebut.edu.cn

Abstract: This study aims to symmetrically improve the economy and environmental protection of combined cooling, heating and power microgrid. Hence, the characteristics of configuration ways of energy storage devices in traditional combined cooling, heating and power systems are analyzed, and a scheme for the operator to establish an energy storage station is designed. An improved aquila optimizer for the optimal configuration of the system is proposed to symmetrically enhance the economic and environmental protection performance. The feasibility of the proposed scheme is verified through experiments in three different places. The results show that the economic cost and exhaust emission of the system with energy storage station are reduced to varying degrees compared with the system with energy storage equipment alone and the system without energy storage equipment based on symmetry concept. Especially in Place 1, the scheme with energy storage station in the system can reduce the electric energy purchased from power grid by 43.29% and 61.09%, respectively, compared with other schemes. This study is conducive to promoting the development of clean energy, alleviating the energy crisis, reducing the power supply pressure of power grid, and improving the profits of operators by symmetrically considering the economic and environmental performance of the system.

Keywords: combined cooling; heating and power system; optimal configuration; energy storage station; improved aquila optimizer; clean energy

Citation: Ning, N.; Liu, Y.-W.; Yang, H.-Y.; Li, L.-L. Design and Optimization of Combined Cooling, Heating, and Power Microgrid with Energy Storage Station Service. *Symmetry* **2022**, *14*, 791. https://doi.org/10.3390/sym14040791

Academic Editor: Peng-Yeng Yin

Received: 24 March 2022
Accepted: 9 April 2022
Published: 11 April 2022

1. Introduction

With the increasingly serious energy crisis and environmental problems, the traditional distributed energy system is gradually being replaced by the integrated energy system, which will become the first choice for the future energy system [1]. The combined cooling, heating, and power (CCHP) system is a typical integrated energy system, and the use of this operating structure can improve the energy utilization efficiency of the system to more than 85% [2]. Heat storage devices can improve the utilization rate of waste heat [3]. Adding renewable energy generation methods, such as photovoltaic power generation and wind power generation, to the traditional CCHP system can improve the environmental protection of the CCHP system and reduce the dependence of the system on non-renewable energy. Increasing the use of clean energy helps reduce carbon emissions [4]. Rising wind power penetration can threaten the stability of the power system [5]. Compared to wind power, photovoltaic power is highly predictable and stable [6,7]. The introduction of photovoltaic/thermal systems can reduce pollutant gas emissions of the combined heat and power system and improve economic efficiency [8]. However, due to the influence of weather factors by temperature, solar irradiance, wind speed, etc., CCHP systems with renewable energy generation methods have uncertainty in electricity production, which will increase the dependence on the grid [9]. The addition of energy storage batteries can alleviate the pressure on the grid [10]. The way that the user installs energy storage batteries in the CCHP system alone will increase the economic cost of the system, and it will also have little effect on reducing the system's dependence on the grid. Therefore, the research on the construction method and capacity configuration of energy storage devices is of great value [11]. There are many variables and constraints involved in the CCHP system, and the uncertainty of the load will also affect the optimal configuration and operation of the system. Consequently, it is relatively difficult to determine the optimal operation of the CCHP system [12,13]. The reasonable construction method of energy storage devices and the optimal configuration of the CCHP system can help the further promotion and application of integrated energy systems.

The various devices in the CCHP system determine how the energy is converted. Currently, CCHP systems are mainly divided into two categories: CCHP systems without energy storage equipment and CCHP systems of self-built energy storage equipment by users themselves [14,15]. Both types of CCHP systems can meet the needs of users, but they also have their own problems. During the peak periods of electricity consumption, CCHP systems without energy storage equipment can only be supplemented by the power grid, which increases the power supply pressure of the power grid [16]. Although the CCHP system of self-built energy storage equipment can reduce the dependence of the system on the power grid, the economic cost of the system itself will increase, and the optimization of the entire system will become more complex [17]. By combining the characteristics of the two systems, this study proposes a multi-microgrid operation method based on energy storage station (ESS) services. Operators establish ESS and take advantage of the scale effects of ESS to serve CCHP systems with different load requirements. While reducing the investment cost of the CCHP system, ESS can make profits by charging electricity service fees, so as to achieve a win-win effect between ESS and CCHP systems. Dealing with the coordinated operation between ESS and CCHP systems is a typical system optimization configuration problem.

Methods for optimal system configuration typically include mathematical programming and intelligent optimization algorithms [18]. Intelligent optimization algorithms can solve objective functions quickly and accurately without changing the system model [19]. Intelligent optimization algorithms have greater advantages in dealing with multi-variable, multi-constraint, and nonlinear problems such as system optimization configuration. In previous studies, classical and emerging algorithms, such as particle swarm optimization (PSO), genetic algorithm (GA), and salp swarm algorithm (SSA), have been applied to the optimal configuration of CCHP systems and achieved considerable results [20–22]. The aquila optimizer (AO) algorithm is a heuristic intelligent optimization algorithm proposed in 2021. Its performance is superior to the traditional PSO algorithm, GA, and novel algorithms such as SSA [23]. Similar to other intelligent optimization algorithms, the AO algorithm also has the problem of slow convergence and tendency to fall into local extremes when dealing with complex problems, so the AO algorithm needs to be improved [24]. The chaos strategy increases the randomness of individuals during the initialization stage, the mutation strategy increases the diversity of the population during the iteration, and the levy flight strategy can prevent individuals from falling into local extremes [25,26]. Therefore, by introducing the above strategies, the improved aquila optimizer (IAO) algorithm is proposed. In this study, the IAO algorithm is applied as the optimization algorithm, and the comparison results are better than those of the compared algorithms.

In this study, we analyze the form of energy storage battery configuration of traditional CCHP system and innovatively propose a storage battery configuration scheme. By introducing the improved strategies into the original AO algorithm, the IAO algorithm is proposed and used for the optimal configuration of the CCHP microgrid model. The addition of photovoltaic power generation equipment improves the sustainability and environmental friendliness of the CCHP microgrid. The proposed method reduces the power supply pressure of the grid, improves the profits of operators, and is conducive to promoting the development of clean energy, alleviating the energy crisis.

The research hypothesis of this study is to reduce the economic cost and exhaust emissions of the CCHP system through optimizing the configuration of the CCHP multi-microgrid system based on ESS service. Through the establishment of ESS by operators, the dependence of the CCHP system on the grid decreases, and the operator makes a profit by providing electrical energy services to the CCHP system. The main work and contributions of this study can be summarized as follows: (1) establishing an optimal configuration model symmetrically considering the economic and environmental benefits of ESS and the CCHP system; (2) proposing the IAO algorithm for optimal system configuration based on the AO algorithm; (3) the proposed algorithm has better optimization performance compared to the original AO algorithm; (4) reducing the dependence of the CCHP system on the grid by establishing ESS; (5) a new energy storage configuration scheme is proposed, which is beneficial to the economic and environmental protection and stable operation of the CCHP system.

The rest of this study is organized as follows: Section 2 briefly analyzes the literature related to the design and optimal configuration of CCHP systems. Section 3 describes the system model proposed in this study. Section 4 describes the original algorithm and the improvement process. Section 5 conducts an actual case study. Section 6 discusses the implications of this study. Finally, conclusions are provided.

2. Literature Review

Conventional power plants are used at about 30% efficiency for fuel energy, with most of the remaining energy lost in the form of waste heat [27]. After more than a century of development, the CCHP system has been proven to be an efficient and energy-efficient device operation mode [28]. The advancement of technology has gradually changed the fuel source and energy circulation mode of CCHP systems. Nami et al. [29] established a model of the solar-assisted biomass energy CCHP system, and analyzed the key factors of system performance optimization. The results show that the energy provided by the system can meet the energy demand of household users. However, the system's technological maturity is low, and the complexity is too high. Moreover, the literature does not evaluate the economics of the system. Wei et al. [30] proposed a CCHP system based on proton exchange membrane fuel cells, optimized the system from the aspects of economy and environmental protection by using the improved mayfly optimizer, and obtained the system configuration that is more economical and environmentally friendly than the original algorithm. However, the system lacks clean energy and the optimization process does not consider the problem of environmental friendliness. Tonekaboni et al. [31] applied nanofluid and porous media to a solar collector and used the collector in a CCHP system to improve the energy absorption efficiency of the system. However, the literature lacks a discussion of the economics and environmental protection of the system. Li et al. [32] used the heat pump as the heat source of the CCHP system. Through the analysis of the primary energy saving rate, the carbon dioxide emission reduction rate, and the annual cost saving rate, the system was optimized and analyzed in practical application. However, this system lacks energy storage equipment, which will cause waste of energy. Nazari et al. [33] investigated the effect of electric boiler, hydro storage, and heat storage tank on the scheduling problem of conventional trigeneration system. The results indicated that the use of the model could improve the profits of the system. However, this system does not contain renewable energy, and the environmental protection of the system needs to be improved. Additionally, this system does not contain energy storage devices, and the lack of energy storage devices can cause energy waste. Previous studies have optimized the design of traditional CCHP systems from the perspective of system energy source, equipment material selection, and energy circulation mode. However, due to the difficulty of promoting and applying new materials, the high economic cost of installation, and the instability of new energy sources, the above-mentioned CCHP systems are still in the theoretical research stage.

The energy crisis and environmental issues have promoted the development of clean and renewable energy, and CCHP systems with clean and renewable energy have become a new trend in recent research [34]. Leonzio et al. [35] used software to design and simulate a trigeneration system powered by biogas. The results showed that the system has significant improvements in power generation and carbon dioxide emissions relative to conventional CCHP systems. However, the application scenarios of biogas power generation methods are limited to rural areas and other areas, and large-scale popularization applications cannot be carried out. Dong et al. [36] designed a CCHP system including wind power generation and photovoltaic power generation equipment as the coupling hub of the electric-gas system. Although this system is more economical than separated systems, the cost of wind turbines installation is still too high for the ordinary user, and the wind turbines are often far away from the user side, which is contrary to the installation concept of CCHP systems integrated on the user side. Zhang et al. [37] used a multi-objective particle swarm algorithm to optimize the CCHP system containing photovoltaic power generation, so that different buildings can achieve a balance between economy and environmental protection. Compared with wind power plants, photovoltaic power generation units can be better used in most scenarios, especially in areas where large equipment and power supply from the grid cannot be installed due to geographical constraints [38]. As a relatively mature clean power generation method, photovoltaic power generation is applied to the CCHP system model built in this study. Photovoltaic power generation equipment often needs to be used in conjunction with energy storage devices to alleviate the intermittent characteristics of power generation [39]. However, in previous studies, energy storage devices were generally configured by the user side of the CCHP system [40]. Considering factors such as installation costs and supporting management, the capacity configuration of the energy storage device is smaller, and it plays a less influential role in buffering and relieving the pressure on the grid. This study proposes that using the ESS operating mode, the operator can establish a large capacity of energy storage devices to serve the CCHP system. The CCHP system needs to pay only the service fee in exchange for the power usage and storage rights of the ESS.

The establishment of ESS increases the difficulty of optimizing the configuration of the CCHP system. Mathematical programming and intelligent optimization algorithms are two commonly used methods for optimizing the configuration. Compared to mathematical programming methods, the use of intelligent optimization algorithms does not require extensive mathematical proofs or changing nonlinear terms in the original model [41]. In the previous studies, GA and PSO have often been used in the optimal configuration and scheduling of the system [42,43]. However, the original algorithm has unstable performance and is easy to fall into local extremums, which has spawned scholars to improve the algorithm and produce new algorithms. Wang et al. [44] introduced the chaos and elite strategies into the original PSO algorithm, which improved the search range and search ability of the original algorithm. Cao et al. [2] replaced the random parameter β in the original owl search algorithm with a circular mapping based on the chaotic mechanism to improve the premature convergence problem of the original algorithm. Abualigah et al. [23] proposed the AO algorithm by simulating the process of finding, tracking, and capturing prey by the aquila. The AO algorithm has been applied to optimize scheduling, parameter tuning, and feature extraction, and has shown superior performance to other algorithms [45,46]. However, when the AO algorithm deals with the multi-variable and multidimensional optimal configuration model proposed in this study, there are still problems of slow convergence speed and low convergence accuracy similar to other algorithms. Therefore, this study proposes the IAO algorithm by improving the different stages of the original AO algorithm.

3. System Description

The structure of the CCHP multi-microgrid system combined with ESS is shown in Figure 1. In this study, the number of the CCHP microgrid is three. Considering the relevant policy restrictions and technical demands, the CCHP system is stipulated not to sell electricity to the grid. The system can be divided into power supply equipment, heating equipment, and cooling equipment. Among them, the power supply equipment includes photovoltaic generator unit (PV), micro turbine (MT), and energy storage station (ESS). Heating equipment includes gas boiler (GB), heat recovery (HR), heat exchanger (HE), and thermal storage tank (TST). Cooling equipment includes electric chiller (EC) and absorption chiller (AC). The mathematical model of the main equipment of the system is described below.

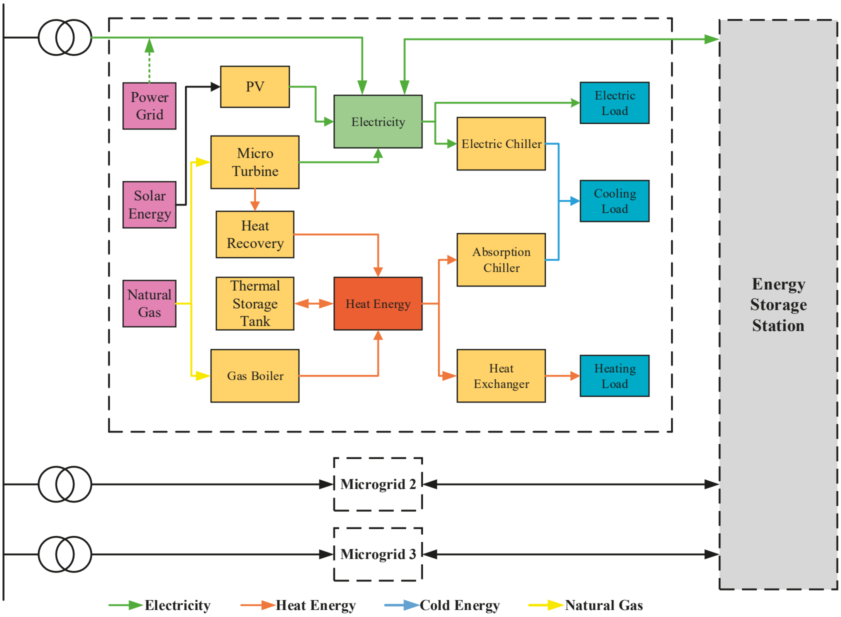


Figure 1. The structure of the CCHP multi-microgrid system combined with ESS.

3.1. Equipment Description

3.1.1. Power Supply Equipment

(1) Photovoltaic power generation model.

Photovoltaic power generation is introduced into the CCHP system as a clean power generation method. It converts solar energy into electrical energy through the photovoltaic effect. The PV output electrical energy is expressed as follows:

$$P_{pv} = N_{pv} \cdot (\frac{G_{pv}}{G_{\text{STC}}}) \cdot [1 + k(T_{pv} - T_{\text{STC}})] \tag{1}$$

where, P_{pv} denotes the power output of PV, N_{pv} denotes the installed capacity of PV, G_{pv} and T_{pv} denote the solar irradiation intensity received by photovoltaic panels and the surface temperature of photovoltaic panels, G_{STC} and T_{STC} denote the solar irradiation intensity received by photovoltaic panels and the surface temperature of photovoltaic panels under standard test conditions, taking values of 1000 W/m^2 and 25 °C, and k denotes the correlation coefficient.

(2) Micro turbine model.

MT is not only responsible for supplying electricity to the load, but the waste heat generated will also be further utilized through HR. The expression of MT output electricity is as follows:

$$P_{mt} = F_{mt} \cdot \alpha \cdot \eta_{mt} \tag{2}$$

where P_{mt} denotes the electrical power output of MT, F_{mt} denotes the consumption volume of natural gas, α is the calorific value of natural gas with a value of 9.7 kWh/m^3, and η_{mt} denotes the power generation efficiency of MT.

(3) Energy storage station model.

The energy storage station is one of the crucial devices in the system. The regulating effect of ESS further improves the performance of the system. Meanwhile, ESS operators can make a profit.

$$P_{bat}^{t} = P_{bat}^{t-1} \cdot (1 - \eta_{bat,loss}) + (P_{buy}^{t} \cdot \eta_{bat,ch} - \frac{P_{sell}^{t}}{\eta_{bat,disch}}) \tag{3}$$

where P_{bat}^{t} and P_{bat}^{t-1} denote the power of ESS at time t and $t-1$, $\eta_{bat,loss}$ denotes the self-discharge rate of ESS, P_{buy}^{t} and P_{sell}^{t} denote the power purchased and sold by ESS from the CCHP system at time t, and $\eta_{bat,ch}$ and $\eta_{bat,disch}$ denote the charging efficiency and discharging efficiency of the energy storage battery.

3.1.2. Heating Equipment

(1) Gas boiler model.

GB plays a crucial role as the final guarantee of heating energy. GB will produce heat when the rest of the equipment cannot meet the heat demand of the user. The expression for its heat generation is as follows:

$$H_{gb} = F_{gb} \cdot \alpha \cdot \eta_{gb} \tag{4}$$

where H_{gb} denotes the heat generated by GB, F_{gb} denotes the consumption volume of natural gas, and η_{gb} denotes the heat production efficiency of GB.

(2) Heat recovery model.

HR makes full use of the fuel consumption of MT. It recovers the waste heat generated by MT in the process of producing electricity. The mathematical expression of HR is as follows:

$$H_{hr} = P_{mt} \cdot COP_{mt} \tag{5}$$

where H_{hr} denotes the heat recovered by HR from MT and COP_{mt} denotes the correlation coefficient of heat production by MT.

(3) Heat exchanger model.

HE connects the heat production of the system with the heat load. The mathematical expression for HE is as follows:

$$H_{he} = \frac{H_{load}}{\eta_{he}} \tag{6}$$

where H_{he} denotes the heat required by HE, H_{load} denotes the heat load demand of the user, and η_{he} denotes the efficiency of HE.

(4) Thermal storage tank model.

TST is a buffering device for heat. TST is preferentially used to store and release excess heat and lack of heat of the system. The state expression of TST is as follows:

$$H_{tst}^{t} = H_{tst}^{t-1} \cdot (1 - \eta_{tst,loss}) + (H_{tst,ch}^{t} \cdot \eta_{tst,ch} - \frac{H_{tst,disch}^{t}}{\eta_{tst,disch}}) \tag{7}$$

where H_{tst}^{t} and H_{tst}^{t-1} denote the heat of TST at time t and $t-1$, $\eta_{tst,loss}$ denotes the self-release rate of TST, $H_{tst,ch}^{t}$ and $H_{tst,disch}^{t}$ denote the heat stored or released at time t, and $\eta_{tst,ch}$ and $\eta_{tst,disch}$ denote the efficiency of TST in storing and releasing heat, respectively.

3.1.3. Cooling Equipment

AC and EC produce cooling to meet the demand of the user. Their mathematical expressions are shown as follows:

$$C_{ac} = H_{ac} \cdot \eta_{ac} \tag{8}$$

$$C_{ec} = P_{ec} \cdot COP_{ec} \tag{9}$$

where C_{ac} and C_{ec} denote the cold generated by AC and EC, H_{ac} denotes the heat consumption of AC, η_{ac} denotes the cooling efficiency of AC, P_{ec} denotes the power consumption of EC, and COP_{ec} denotes the cooling coefficient of EC.

3.2. Problem Description

This study aims to optimize the configuration of the CCHP multi-microgrid system combined with ESS. In the solution process, both the economic and environmental protection of the CCHP system and the profitability of the ESS should be satisfied. The decision variables, objective function, constraints, and operating scheme are described as follows.

3.2.1. Decision Variables

The decision variables are as follows:

$$X = [N_{pv}, N_{mt}, N_{gb}, N_{tst}, N_{grid}, N_{bat}, r] \tag{10}$$

where N_{pv}, N_{mt}, N_{gb}, and N_{tst} denote the installed capacity of PV, MT, GB, and TST, N_{grid} denotes the upper limit of power purchased from the grid, N_{bat} denotes the capacity of storage batteries to be allocated, and r denotes the ratio of EC cooling production to the cooling load.

3.2.2. Objective Function

The objective function considers the net investment cost of the CCHP system, the cost of consumed natural gas, the ESS service charge, the cost of electricity purchased from the grid, and the cost of waste gas treatment.

$$F = C_I + C_F + C_S + C_G + C_E \tag{11}$$

where F denotes the value of the objective function, C_I denotes the daily investment cost of the equipment, C_F denotes the cost of consuming natural gas, C_S denotes the service cost of ESS, C_G denotes the cost of purchasing electricity from the grid, and C_E denotes the cost of treating the waste gas. The specific cost expressions are as follows:

$$C_I = \frac{\beta \cdot \sum\limits_{i=1}^{n} N_i \cdot C_i}{365 \cdot L} \tag{12}$$

$$\beta = \frac{a \cdot (1+a)^L}{(1+a)^L - 1} \tag{13}$$

where β denotes the investment recovery coefficient, N_i denotes the installed capacity of the i-th equipment, C_i denotes the unit investment cost, L denotes the life of the equipment, and a denotes the discount rate, taking the value of 0.08.

$$C_F = (F_{mt} + F_{gb}) \cdot C_f \tag{14}$$

$$C_G = \sum_{t=1}^{24} P_{buy,grid}^{t} \cdot C_e^t \tag{15}$$

where C_f denotes the price of natural gas, $P^t_{buy,grid}$ denotes the electricity purchases from the grid at time t, and C^t_e is the price of electricity at time t.

$$C_S = \sum_{t=1}^{24} [(P^t_{sell} \cdot C^t_{sell} - P^t_{buy} \cdot C^t_{buy}) + (P^t_{sell} + P^t_{buy}) \cdot C_{serve}] \tag{16}$$

where P^t_{sell} and P^t_{buy} represent the electricity sold and bought by ESS to the CCHP system at time t, C^t_{sell} and C^t_{buy} represent the prices of electricity sold and purchased by ESS at time t, and C_{serve} represents the service charge of ESS taking the value of 0.0079 \$/kWh.

$$C_E = C_{GT} + \sum_{t=1}^{24} \lambda(P^t_{waste} + H^t_{waste}) \tag{17}$$

$$C_{GT} = \sum_{t=1}^{24} \sum_{g=1}^{3} (P^t_{mt} \cdot \gamma^{mt}_g + H^t_{gb} \cdot \gamma^{gb}_g + P^t_{buy,grid} \cdot \gamma^{grid}_g) \cdot \beta_g \tag{18}$$

where P^t_{waste} and H^t_{waste} are the electricity and heat waste at time t, λ is the penalty factor, P^t_{mt}, H^t_{gb}, and $P^t_{buy,grid}$ are the electricity generated by MT, the heat produced by GB, and the electricity purchased from the grid at time t, γ^{mt}_g, γ^{gb}_g, and γ^{grid}_g are the emissions of the g-th pollutant gas emitted by MT, GB, and the grid, and β_g denotes the cost required to treat the g-th pollutant gas.

3.2.3. Constraints

The constraints include the balance of electricity, heating and cooling, and the capacity limits of the equipment.

(1) Equality constraints.

The electricity balance is as follows:

$$P^t_{pv} + P^t_{mt} + P^t_{buy,grid} + P^t_{sell} + P^t_{vacancy} = P^t_{load} + P^t_{ec} + P^t_{buy} + P^t_{waste} \tag{19}$$

where P^t_{load} indicates the required electricity of the system at time t, and $P^t_{vacancy}$ and P^t_{waste} are the shortage and waste of electricity at time t.

The heating balance is as follows:

$$H^t_{hr} + H^t_{tst,disch} + H^t_{gb} + H^t_{vacancy} = H^t_{load} + H^t_{tst,ch} + H^t_{waste} \tag{20}$$

where $H^t_{vacancy}$ and H^t_{waste} represent the shortage and waste of heat at time t, respectively.

The cooling balance is as follows:

$$C^t_{ac} + C^t_{ec} = C^t_{load} \tag{21}$$

where C^t_{load} denotes the cooling load required by the user at time t.

(2) Inequality constraints.

The inequality constraints are mainly the capacity limits of the equipment. The inequality constraints involved in this study are listed in Table 1.

Table 1. Inequality constraints of the system.

Equipment	Value (kW)
PV capacity	(0, 600]
MT capacity	(0, 500)
GB capacity	(0, 400)
TST capacity	(0, 300]
GRID limit	(0, 400]
Charge and discharge limit of battery	$[0, 0.5N_{bat}]$
Charge and discharge limit of TST	$[0, 0.4N_{tst}]$

3.3. Operation Schemes

The addition of storage batteries improves the utilization of electricity. However, the configuration of storage batteries alone will increase the investment cost. Additionally, the smaller configuration size has less effect on electricity regulation. Therefore, this study proposes a CCHP system based on ESS service. The earnings of ESS come from two sources. First, the electricity used and stored in ESS is calculated for each CCHP system: charging in the form of electricity sales and purchases. Second, ESS charges a service fee for the electricity used and stored by the CCHP system. In this study, two other operation schemes are chosen for comparison. The three operation schemes are described as follows.

Scheme 1: ESS service model

Each CCHP system operates independently without configuring energy storage batteries, and the ESS services the CCHP system. Excess power is sold to ESS when PV and MT generation is sufficient. Conversely, electricity is first purchased from ESS when there is insufficient electricity. If the ESS does not have enough electricity, the customer purchases the shortage from the grid.

Scheme 2: Users configure energy storage equipment by themselves

Each CCHP system operates independently and is equipped with energy storage batteries. The operation process is similar to scheme 1. However, under this scheme, the cost of energy storage batteries needs to be included in the net investment cost.

Scheme 3: CCHP system is not equipped with energy storage equipment

Each CCHP system operates independently without the configuration of energy storage batteries. When PV and MT generation is sufficient, surplus electricity is wasted. Conversely, electricity is purchased from the grid if there is insufficient generation.

The heating and cooling load demands are met in the same way for the three operating schemes. First, it is determined whether the waste heat recovered by HR meets the system heat demand. If the waste heat is larger than the demand, the excess heat is saved in the TST. If the waste heat is insufficient, TST is given priority for heat release. When TST still cannot meet the demand, GB starts heat production.

4. Optimization Algorithm

This section gives a detailed description of the original AO algorithm and its improvement process.

4.1. AO Algorithm

The aquila optimizer is a novel intelligent optimization algorithm based on the predation process of the aquila [17]. According to the different flight behaviors of the aquila, the predation process includes two stages: searching for prey and catching prey. The exact procedure of the original algorithm is as follows.

4.1.1. Search Stage

When the number of iterations is less than two-thirds of the maximum iterations, the aquila individual updates the position by Equation (22) or Equation (24). The specific position update equation is determined by judging the size of the random number. The search stage is as follows:

$$X_1(t+1) = X_{best}(t) \times (1 - \frac{t}{T}) + (X_M(t) - X_{best}(t) * rand) \tag{22}$$

$$X_M(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{23}$$

where $X_1(t + 1)$ denotes the position of the individual after the update, t denotes the current iteration number, $X_{best}(t)$ denotes the position of the best individual, $X_M(t)$ denotes the average of all individual positions, T denotes the maximum iteration number, N is the number of individuals, and *rand* is a random value between 0 and 1.

$$X_2(t+1) = X_{best}(t) \times Levy(D) + X_R(t) + (y - x) * rand \tag{24}$$

where $X_2(t + 1)$ denotes the position of the individual after the update, $X_R(t)$ denotes the position of a random individual in the current population, D denotes the dimension of the search space, *Levy* denotes the levy flight function, and y and x are used to represent the spiral process of the aquila. The corresponding mathematical expressions are as follows:

$$Levy(D) = 0.01 \times \frac{u \times \delta}{|v|^{\frac{1}{\beta}}} \tag{25}$$

$$\delta = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right) \tag{26}$$

$$\begin{cases} x = r \times \sin(\theta) \\ y = r \times \cos(\theta) \\ r = r_1 + U \times D_1 \\ \theta = -\omega \times D_1 + \frac{3\pi}{2} \end{cases} \tag{27}$$

where β takes the value of 1.5, u and v are two random numbers taking values between 0 and 1, r_1 denotes the number of search periods and takes values between 1 and 20, U and ω are fixed values of 0.0565 and 0.005, respectively, and D_1 is an integer between 1 and D.

4.1.2. Catch Stage

When the number of iterations is greater than two-thirds of the maximum iterations, the aquila individual updates the position by Equation (28) or Equation (29). Again, the specific position update equation is determined by judging the size of the random number. The catch stage is as follows:

$$X_3(t+1) = (X_{best}(t) - X_M(t) \times \alpha - rand + ((UB - LB) \times rand + LB) \times \delta \tag{28}$$

where $X_3(t + 1)$ denotes the position of the individual after the update, α and δ are adjustment parameters taking fixed values of 0.1, and *UB* and *LB* denote the upper and lower bounds of the search space.

$$X_4(t+1) = QF(t) \times X_{best}(t) - (G_1 \times X(t) \times rand - G_2 \times Levy(D)) + rand \times G_1 \tag{29}$$

$$\begin{cases} QF(t) = t^{\frac{2 \times rand - 1}{(1-T)^2}} \\ G_1 = 2 \times rand - 1 \\ G_2 = 2 \times (1 - \frac{t}{T}) \end{cases} \tag{30}$$

where $X_4(t+1)$ denotes the position of the individual after the update, $QF(t)$ denotes the search function, G_1 denotes the movement parameter, and G_2 denotes the flight slope.

4.2. IAO Algorithm

The original performance of the AO algorithm is better than most intelligent algorithms already. However, similar to other heuristic algorithms, the AO algorithm still has the problem of slow convergence and the tendency to fall into local optimum. The AO algorithm is improved to increase the convergence speed and convergence accuracy. First, the logistic chaotic mapping is used for the initialization process of the population. Thus, the initial distribution of individuals in the aquila population is improved. Second, mutation, hybridization, and competition strategies from the differential evolution algorithm are introduced into the population update to improve the diversity of the population. Finally, the aquila in the optimal position will execute the levy flight strategy, thus preventing it from falling into local extremes.

4.2.1. Logistic Chaos Mapping Strategy

The initialization quality of the population affects the entire population search process. In the initial stage, a more random initial distribution enables individuals to perform a better search for the global. Individuals have a greater probability of approaching the optimal solution, thus increasing the accuracy and iteration speed of the algorithm. In this study, the logistic chaotic mapping is selected for the initialization process of the population. The specific expressions are as follows:

$$r_i = \mu \times r_{i-1} \times (1 - r_{i-1}), \qquad i \in [2, 3, \ldots, N] \tag{31}$$

$$X_i = r_i \times (UB - LB) + LB \tag{32}$$

where r_i denotes the i-th random value generated by the chaotic mapping, X_i denotes the initialized position of the individual after the chaotic mapping, and UB and LB are the upper and lower bounds of the search space.

4.2.2. Mutation, Hybridization, and Competition Strategies

To improve the diversity of the aquila population during the predation, and thus, to solve the problems caused by the lack of diversity during the search process, mutation, hybridization, and competition strategies are introduced after each iteration of the population. The execution process is as follows:

$$U_i(t) = X_{q1}(t) + F \times [X_{q2}(t) - X_{q3}(t)] \tag{33}$$

$$\begin{cases} F = F_0 \times 2^{\tau} \\ \tau = \exp(1 - \frac{T}{1+T-t}) \end{cases} \tag{34}$$

where $X_{q1}(t)$, $X_{q2}(t)$ and $X_{q3}(t)$ are three different individuals within the population after the t-th iteration, $U_i(t)$ is the new individual position generated by mutation, and F is the dynamic mutation parameter. Then, the new population generated by mutation and the original population are crossed to produce the hybrid population by hybridization. The specific hybridization process is as follows:

$$V_i(t) = \begin{cases} U_i(t) & rand \leq CR \\ X_i(t) & else \end{cases} \tag{35}$$

where $V_i(t)$ denotes the position of the new individual produced by hybridization, and CR is a random parameter taking values between 0.5 and 1. Finally, the new population produced after the mutation and hybridization process competes with the original population to keep the superior individuals. The specific competition process is as follows:

$$X_{new\_i}(t) = \begin{cases} V_i(t) & f[V_i(t)] \leq f[X_i(t)] \\ X_i(t) & else \end{cases} \tag{36}$$

where $X_{new\_i}(t)$ denotes the location of the individual generated after the competition, and f denotes the fitness function, which is the objective function in this study.

Compared with the original population, the positions of all individuals in the new population undergo a larger perturbation. The search range is wider in the early iterative stage, thus avoiding falling into the local optimum. The introduction of the competition strategy retains the better individuals in the population and eliminates the worse ones, which further improves the convergence accuracy of the algorithm.

4.2.3. Levy Flight Strategy

The levy flight strategy is already involved in the position update formulation of the original algorithm, but it is necessary to execute the strategy again for the optimal individual. After all individuals in the population have completed one complete iteration, the current best individual is selected for levy flight. The fitness values of the individual are compared before and after the execution of levy flight strategy to update the optimal individual position. The process is shown as follows:

$$X_{_{new\_best}}(t) = \begin{cases} X_{levy\_best}(t) & f[X_{levy\_best}(t)] \leq f[X_{best}(t)] \\ X_{best}(t) & else \end{cases} \tag{37}$$

$$X_{levy\_best}(t) = X_{best}(t) + 0.05 \times Levy(D) \tag{38}$$

where $X_{best}(t)$ denotes the optimal individual position in the population after the t-th iteration, $X_{levy-best}(t)$ denotes the updated position of the optimal individual position after the levy flight strategy, and $X_{new-best}(t)$ is the optimal individual position after the selection.

Figure 2 shows the flow chart of the optimal configuration of the CCHP system based on the IAO algorithm. The specific steps are as follows:

Step 1 Input load and weather data, and parameters of equipment and the algorithm;

Step 2 Initializing the population with the logistic chaos mapping;

Step 3 Calculate the current position of the individual and the fitness value;

Step 4 The aquila individual enters the search and catch stage;

Step 5 The population obtained from Step 4 performs mutation, hybridization, and competition strategies through Equations (33), (35), and (36) to obtain a new population, thereby retaining the individuals with low fitness value;

Step 6 The optimal individual performs the Levy flight strategy according to Equations (37) and (38);

Step 7 Judge whether the maximum number of iterations is reached, and if not, return to Step 3, otherwise execute the next step;

Step 8 Save data and output the objective function value.

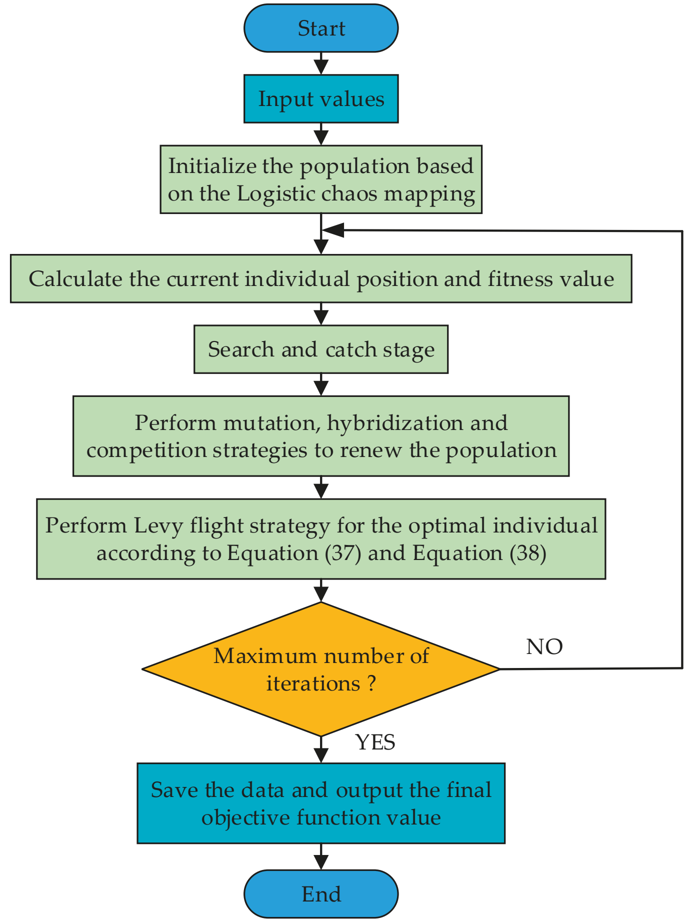

Start
Input values
Initialize the population based on the Logistic chaos mapping
Calculate the current individual position and fitness value
Search and catch stage
Perform mutation, hybridization and competition strategies to renew the population
Perform Levy flight strategy for the optimal individual according to Equation (37) and Equation (38)
Maximum number of iterations ?
NO
YES
Save the data and output the final objective function value
End

Figure 2. The flow chart of optimal configuration based on the IAO.

5. Case Study

To evaluate the effectiveness of the operation scheme and optimization algorithm proposed in this study, the load data of a large hotel, a supermarket, and a primary school in 16 typical buildings provided by the US Department of Energy Building Technologies program are selected as cases for analysis [47,48]. Figure 3 shows the load curves of the three buildings. Figure 4 shows the temperature and solar irradiance curves. Table 2 lists the relevant parameters of the equipment in the system. Table 3 shows the installation cost and life of the equipment. Table 4 shows the price of electricity and natural gas. Table 5 shows the pollutant gas emission factors and treatment costs.

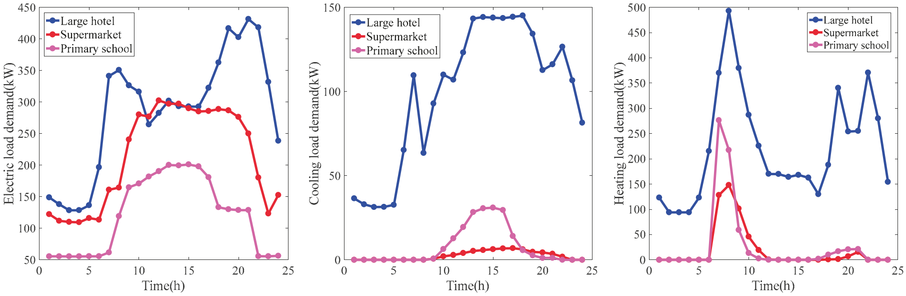


Figure 3. The load demand of the buildings.

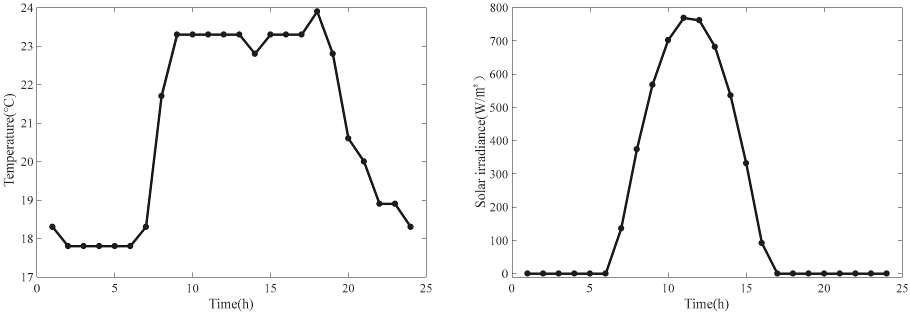


Figure 4. Temperature and solar irradiance curves.

Table 2. The value of the equipment parameters.

Equipment	Parameter	Symbol	Value
Micro turbine	efficiency	η_{mt}	0.2
ESS/Battery	charge/discharge efficiency	$\eta_{bat,ch}$, $\eta_{bat,disch}$	0.95
	self-discharge rate	$\eta_{bat,loss}$	0.02
Gas boiler	efficiency	η_{gb}	0.75
Heat recovery	coefficient of performance	COP_{mt}	1.8
Heat exchanger	efficiency	η_{he}	0.95
Thermal storage tank	charge/discharge efficiency	$\eta_{tst,ch}$, $\eta_{tst,disch}$	0.9
	self-discharge rate	$\eta_{tst,loss}$	0.05
Electric chiller	coefficient of performance	COP_{ec}	3.5
Absorption chiller	efficiency	η_{ac}	0.85

Table 3. The investment cost and the life of the equipment.

Equipment	PV	MT	GB	HR	EC	AC	HE	BAT	TST
Unit price ($/kW)	1996	1432	393	134	260	193	24	228	16
Life (year)	20	20	20	20	20	20	20	10	20

Table 4. The price of electricity and natural gas.

	Time	C_e^t ($/kWh)	P_{sell}^t ($/kWh)	P_{buy}^t ($/kWh)	C_f ($/m^3)
Peak	8:00–12:00, 17:00–21:00	0.2138	0.1808	0.1493	0.0366
Balance	12:00–17:00, 21:00–24:00	0.1289	0.1179	0.0864	
Valley	0:00–8:00	0.0582	0.0629	0.0314	

Table 5. Pollutant gas emission and treatment parameters.

Gas Type	γ_g^{mt} (g/kWh)	γ_g^{gb} (g/kWh)	γ_g^{grid} (g/kWh)	β_g ($/kg)
CO_2	724	254	922	0.033
NOx	0.2	0.54	2.295	9.898
SO_2	0.0036	0.764	3.583	2.333

Note: The definitions of γ_g^{mt}, γ_g^{gb}, γ_g^{grid}, and β_g are consistent with Equation (18).

In this study, the population size of the algorithm is 30, and the number of iterations is 500. The experiments were performed on MATLAB R2016b software. The computer is configured with Intel® Core^{TM} i5-6200U, 2.4 GHz, 12 GB RAM, and is made in China.

5.1. Optimization Results

This study aims to improve the economy and environmental protection of the system by combining ESS. By arranging the energy flow of the CCHP system properly, it can reduce the pressure on the grid and the pollutant gas emissions. Meanwhile, ESS makes a profit by serving the CCHP system. The improvements to the original AO algorithm have improved its performance. The gray wolf optimizer (GWO), the whale optimization algorithm (WOA), and the original AO algorithm are selected as comparison algorithms to verify the performance of the IAO algorithm. Based on the different schemes, four algorithms are used to optimize the system, respectively. Section 3.3 provides a specific description of the schemes. The results are the average values of 50 runs based on the program. The convergence curves of the four algorithms are shown in Figures 5–7.

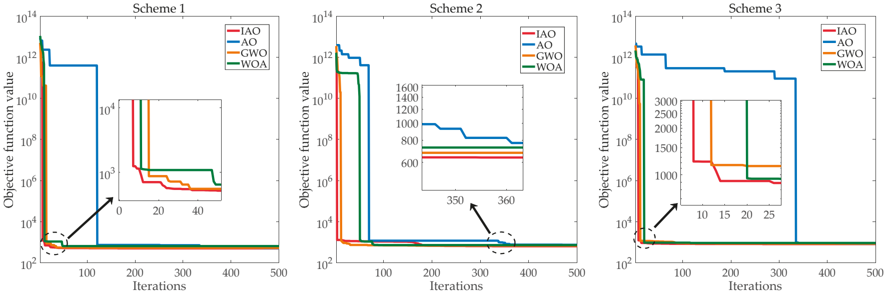


Figure 5. The convergence curves of Place 1 under different schemes.

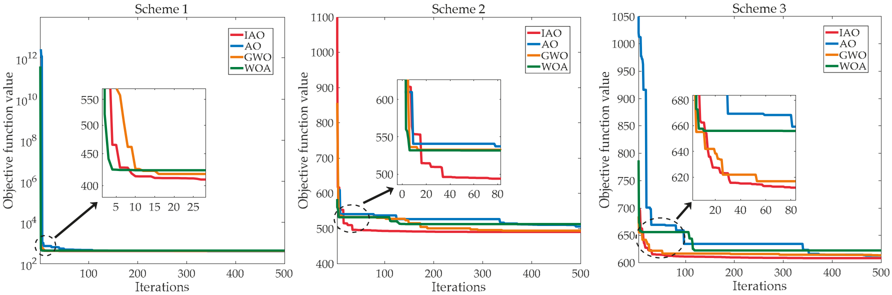


Figure 6. The convergence curves of Place 2 under different schemes.

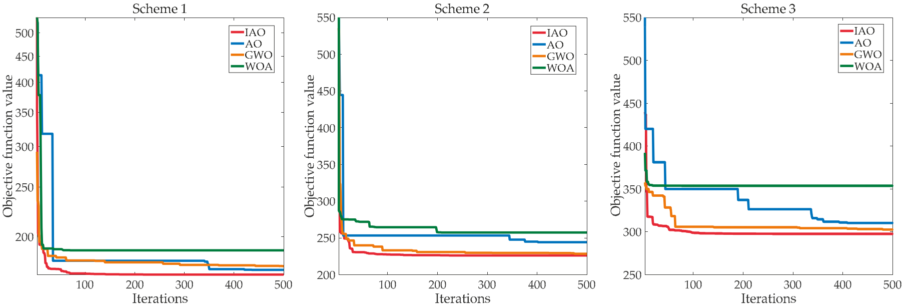


Figure 7. The convergence curves of Place 3 under different schemes.

The analysis of the convergence curves reveals the following conclusions.

(1) The convergence speed of the IAO algorithm is significantly improved compared to the original AO algorithm. At the iteration number of 100, the IAO algorithm can already converge to a smaller value. Moreover, the convergence speed of the IAO algorithm is faster compared with GWO and WOA.

(2) The convergence accuracy of the IAO algorithm is higher. When reaching the maximum number of iterations, the convergence value of the IAO algorithm is smaller than the convergence values of the other three compared algorithms.

(3) For the same place and optimization algorithm, the convergence values of the iterative curves based on different schemes have significant differences. Specifically, the convergence value of the iteration curve is smaller when the system is operating with scheme 1.

The above results show that the improvement to the AO algorithm is effective. The convergence speed and convergence accuracy of the IAO algorithm are better than the comparison algorithms. Meanwhile, the operation scheme proposed in this study has better economy and environmental protection.

Table 6 shows the objective function values and pollutant gas emissions for different operation schemes.

Table 6. Objective function values and pollutant gas emissions.

Scheme	**Algorithm**	**Large Hotel**		**Supermarket**		**Primary School**	
		***F*/$**	***G*/g**	***F*/$**	***G*/g**	***F*/$**	***G*/g**
1	IAO	503.41	3.81×10^6	402.39	1.75×10^6	168.47	8.25×10^5
	AO	522.61	3.85×10^6	413.06	1.80×10^6	180.94	8.37×10^5
	GWO	534.14	3.86×10^6	413.38	1.76×10^6	175.06	8.32×10^5
	WOA	641.90	3.96×10^6	422.74	1.78×10^6	187.91	8.38×10^5
2	IAO	635.55	4.58×10^6	490.45	2.45×10^6	226.01	1.23×10^6
	AO	743.94	5.21×10^6	515.44	2.50×10^6	245.17	1.25×10^6
	GWO	673.85	4.76×10^6	494.12	2.46×10^6	228.41	1.24×10^6
	WOA	727.72	5.02×10^6	512.91	2.47×10^6	257.25	1.27×10^6
3	IAO	810.40	5.64×10^6	608.54	3.09×10^6	297.56	1.63×10^6
	AO	844.37	5.68×10^6	614.71	3.11×10^6	311.26	1.67×10^6
	GWO	850.13	5.80×10^6	613.61	3.11×10^6	302.55	1.66×10^6
	WOA	913.95	5.89×10^6	622.52	3.13×10^6	353.59	1.76×10^6

Note: *F* indicates the objective function value and *G* indicates the pollutant gas emission.

The data in Table 6 more intuitively show the effectiveness of the optimal configuration using the IAO algorithm. For different places and different schemes, the proposed algorithm can obtain the minimum objective function value. Therefore, the IAO algorithm has better search capability and stability than other algorithms. Meanwhile, the calculation shows that the daily economic cost obtained by the IAO algorithm is lower when the systems of the three places are operating with scheme 1. Compared with the cost of scheme 2, the values decreased by 20.54%, 17.95%, and 25.46%, respectively. Compared with the cost when the systems of three places are operating with scheme 3, the values decreased by 37.88%, 33.88%, and 43.38%, respectively. The values for pollutant gas emissions decreased by 16.8%, 28.57%, 32.93% and 32.45%, 43.37%, 49.39%, respectively. The results indicate that the construction of ESS is beneficial to the operation of the CCHP system. Therefore, scheme 1 achieves both economic and environmental improvements for the user side of the CCHP system.

This study aims to take advantage of the scale of ESS to improve the performance of the CCHP system. At the same time, the ESS operator can be profitable. Table 7 shows the capacity configurations and daily investment costs of the energy storage batteries and the revenue of the operator.

Table 7. Energy storage configuration results.

	Scheme 1	Scheme 2		
		Place 1	Place 2	Place 3
Capacity/kWh	2804	431	303	202
Cost/$	26.10	4.01	2.82	1.88
Revenue/$	126.76		——	

Note: The cost is obtained by Equation (12) and the revenue is obtained by Equation (16).

From the data in Table 7, it can be found that compared to the CCHP system in scheme 2, which is configured with energy storage equipment alone, in scheme 1, ESS can be configured with larger capacity energy storage batteries. Compared with scheme 2, although the form of building ESS will increase the investment cost in energy storage equipment, the merchant still has room for profit. Therefore, the configuration of energy storage equipment proposed in this study is feasible. We consider the profitability of the merchant, and also take into account the economy and environmental protection of the CCHP system. A win-win situation can be achieved by operating ESS with multiple CCHP systems.

5.2. Scheduling Results

Taking Place 1 as an example, Figures 8–10 show the electricity balance and heating balance of the CCHP system obtained by the IAO algorithm under different schemes.

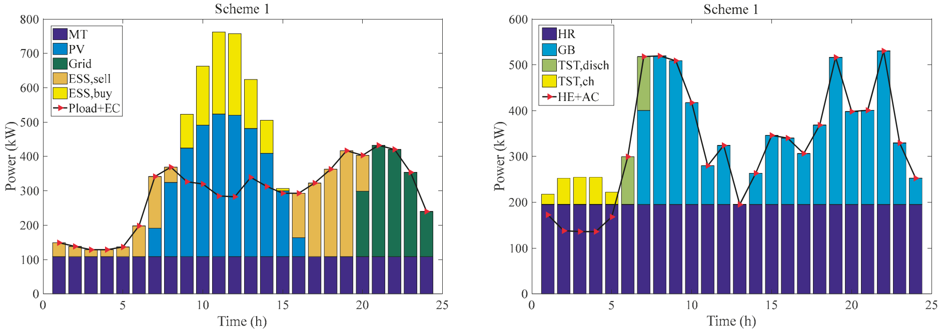


Figure 8. The electricity balance and heating balance under scheme 1.

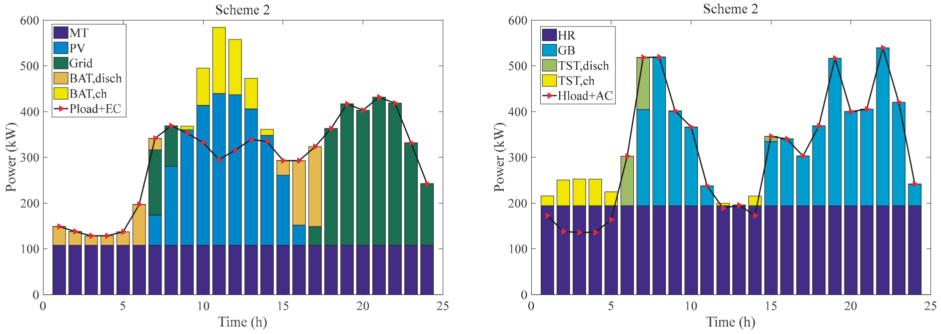


Figure 9. The electricity balance and heating balance under scheme 2.

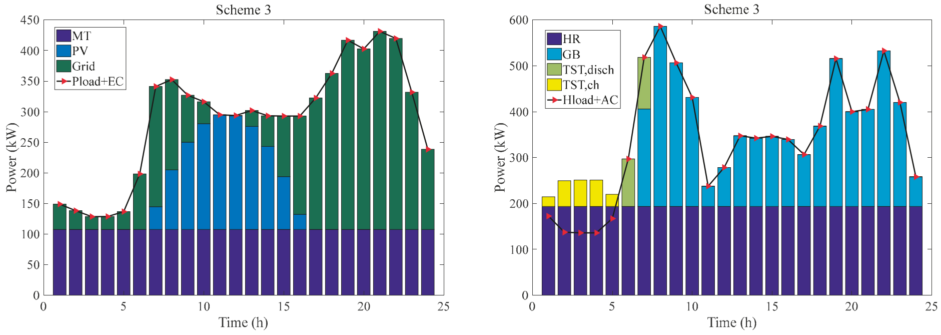


Figure 10. The electricity balance and heating balance under scheme 3.

The electricity balance shows that all three operating schemes can meet the electricity demand of Place 1 during the day. For scheme 1, from 01:00 to 19:00, the ESS regulates the electrical energy balance of the CCHP system by selling and purchasing electricity. ESS and MT work together to meet the electricity demand of the system when weather conditions do not allow for PV generation. When the PV can generate electricity, ESS, MT, and PV together meet the electricity balance of the system. From 20:00 to 24:00, ESS, MT, and the grid work together to meet the electricity demand of the system. For scheme 2, the scale of self-configured energy storage equipment in CCHP systems is smaller. Therefore, its adjustable range for electricity reduces accordingly. As a result, the electricity required from the grid increases. For scheme 3, the lack of energy storage equipment prevents the system from configuring large-capacity photovoltaic power generation equipment. The system still requires a significant amount of electricity from the grid to meet demand. In summary, ESS can guarantee the electricity balance of the CCHP system by taking advantage of its scale. I In addition, it makes a profit by selling electricity and charging service fees. Meanwhile, the CCHP system reduces economic costs by eliminating energy storage equipment and selling surplus electricity.

The heating balance shows that all three operating schemes can meet the heating demand of Place 1 during the day. The heat production from 1:00 to 11:00 and 15:00 to 24:00 is essentially the same for the different equipment under the three schemes. From 12:00 to 14:00, PV produces sufficient electricity. Due to the electricity regulation by energy storage equipment, the system can be configured with higher capacity photovoltaic generation equipment when operating through scheme 1 and scheme 2. As a result, the percentage of the cooling load allocated to EC will increase. Consequently, less heat will be required to meet the cooling load, indirectly reducing the heat production of GB, thus reducing the emission of polluting gases. On the contrary, when the system is operating with scheme 3, the capacity of the configured photovoltaic power generation equipment is less due to the lack of energy storage equipment. Hence, EC receives less electricity and reduces its cooling production, and more heat is needed to meet the demand for AC. More heat needs to be provided by GB, thus increasing the emission of polluting gases.

Table 8 shows the electricity generated by different equipment under the three operating schemes. With Scheme 1 operation, the PV and the grid provide 34.38% and 15.08% of the total electricity. When run with scheme 2, the corresponding values are 29.19% and 28.27%. When run with scheme 3, the corresponding values are 17.92% and 44.76%, respectively. The percentage of electricity provided by energy storage equipment under the three schemes is 17.95%, 8.04%, and 0%, respectively. Therefore, for the same load demand, increasing the proportion of PV can reduce the pressure on the grid to supply electricity. Compared with scheme 2 and scheme 3, the electricity provided by the grid is decreased by 43.29% and 61.09% when operating with scheme 1. The intermittency of PV needs to be balanced by energy storage equipment. ESS purchases excess electricity during the peak generation period of PV, and sells electricity preferentially when the system lacks electricity. The combined model of ESS and PV produces almost zero pollution in the production and utilization of electricity. The scheme also makes more sense for environmental protection.

Table 8. The electricity production of the equipment.

Scheme	MT/kWh	PV/kWh	ESS (Battery)/kWh	Grid/kWh	Total/kWh
1	2602	2745	1433	1204	7984
2	2590	2192	604	2123	7509
3	2580	1239	—	3094	6913

In summary, the CCHP system based on ESS service proposed in this study has more competitive advantages. The system is more economical and emits fewer pollutant gases. Moreover, ESS has room for profitability. The energy storage space provided by ESS can better alleviate the intermittency of PV, regulate the electricity balance of the system, and reduce the pressure on the grid.

6. Discussion

Compared with the original AO algorithm, the improved algorithm improves the convergence speed and convergence accuracy and can obtain better objective function values in fewer iterations. Based on the better objective function value, it can be concluded that the economic cost of the system is reduced and the exhaust emissions are reduced. This study contributes to the optimal configuration of the CCHP system. The optimal configuration model based on the IAO algorithm has played an essential role in influencing and promoting the related fields.

In terms of reducing economic costs, this study proposes a new energy storage equipment configuration scheme. Through the coordination of energy storage equipment and photovoltaic power generation equipment, the demand of the system for grid power supply is reduced. Therefore, the exhaust gas emissions from grid power generation are reduced, thus symmetrically reducing the economic investment required for exhaust gas treatment. Meanwhile, the reduced power supply can reduce the pressure on the grid and improve the flexibility of the CCHP system. The CCHP system can still maintain a relatively stable operation when an accident occurs on the grid. In addition, through the construction of ESS, the intermittency of photovoltaic power generation can be further alleviated, and the ESS operator can still get the revenue to keep operating normally.

In terms of exhaust emissions, the system emits the least quantity of exhaust gases when operating with the scheme proposed in this study. The quantity of the exhaust emissions is linked to the cost required to treat the exhaust gas. In addition, the reduction of exhaust emissions can also decrease the damage to the atmosphere. With lower emissions, the cost of treating the exhaust gas is reduced, thus saving the economic investment of the system. As an important part of the ecological environment, reducing exhaust gas pollution and maintaining a clean atmosphere can promote the green cycle of the entire ecosystem. This study provides a sustainable and effective new idea for the green development of the microgrid.

This study makes full use of clean energy to reduce the consumption of non-renewable energy sources. Under the premise of meeting the load requirements from users, this study is committed to finding an effective system configuration solution. In the context of energy crisis and environmental problems, energy conservation and emission reduction are important issues that both companies and individuals cannot ignore. The CCHP system has raised the utilization rate of energy to a new level. The emergence of CCHP systems provides an effective implementation way for energy cascade utilization. Based on the traditional CCHP system, the introduction of photovoltaic power generation equipment can improve the environmental protection benefits of the system and save fossil fuels. Considering the current rising prices of non-renewable energy, reducing the consumption of natural gas raw materials and making full use of clean power generation methods can reduce dependence on imported energy. This study provides an effective way to reduce energy consumption and reduce carbon emissions, which can promote sustainable development.

7. Concluding Remarks

This study designed a CCHP system including photovoltaic power generation equipment and energy storage equipment. According to the characteristics of the two typical configuration schemes of energy storage equipment, the operation scheme based on ESS service was proposed, and a mathematical model considering economy and environmental protection was established. Additionally, the IAO algorithm was proposed by introducing improved strategies. The optimal configurations of the system under different operating schemes were obtained by the IAO algorithm and other compared algorithms. The conclusions are as follows.

(1) The IAO algorithm has better seeking ability than the original algorithm and other compared algorithms. The IAO algorithm has a faster convergence speed and higher convergence accuracy. When other conditions are consistent, using the IAO algorithm to optimize the system can get better results.

(2) The operating scheme proposed in this study is more feasible. For the system in the same place when operating with three different schemes, scheme 1 shows a greater advantage in terms of economy and environmental protection. Compared with scheme 2 and scheme 3, the daily economic cost of the three places decreased by 20.54%, 17.95%, 25.46% and 37.88%, 33.88%, 43.38%, respectively, and the pollutant gas emissions decreased by 16.8%, 28.57%, 32.93% and 32.45%, 43.37%, 49.39% respectively.

(3) Whether to configure energy storage equipment and how to configure energy storage equipment directly affect the configured capacity of PV in CCHP microgrid, thus affecting the pressure of power grid supporting system. Although the three operation schemes can meet the load demand of the system, their economy is different. Scheme 1 can configure a larger scale of PV to reduce the power supply pressure of power grid by matching the PV with the ESS. Taking Place 1 as an example, the electricity provided by the grid is decreased by 43.29% and 61.09%, respectively, when operating with scheme 1 compared to scheme 2 and scheme 3, which improves the system's economic performance.

We put forward a new scheme for the energy storage configuration of CCHP system, and propose the IAO algorithm to optimize the system. The proposed method reduces the power supply pressure of the grid, improves the profits of operators, and is conducive to promoting the development of clean energy, alleviating the energy crisis. However, this study has some limitations. There is only one renewable energy in the system, namely photovoltaic energy, and more kinds of renewable energy can be taken into account in future research.

Author Contributions: Conceptualization, N.N. and Y.-W.L.; methodology, Y.-W.L. and L.-L.L.; software N.N. and H.-Y.Y.; writing—original draft preparation, N.N. and Y.-W.L.; writing—review and editing, N.N. and Y.-W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the key project of the Tianjin Natural Science Foundation (Project No. 19JCZDJC32100) and the Natural Science Foundation of Hebei Province of China (Project No. E2018202282).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not Applicable.

Acknowledgments: The authors thank Bimenyimana Samuel for valuable help in the final version of this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

AC	absorption chiller
AO	aquila optimizer
bat	battery
CCHP	combined cooling, heating and power
ch	charge
COP	coefficient of performance
disch	discharge
EC	electric chiller
ESS	energy storage station
f	fitness function
F	objective function value
G	pollutant gas emission
GA	genetic algorithm
GB	gas boiler
GWO	gray wolf optimizer
H	heat energy
HE	heat exchanger
HR	heat recovery
IAO	improved aquila optimizer
MT	micro turbine
P	power
PSO	particle swarm optimization
PV	photovoltaic generator unit
SSA	salp swarm algorithm
TST	thermal storage tank
WOA	whale optimization algorithm
η	efficiency

References

1. Zhang, Y.; Jiang, J.; Zhang, X.; Sun, L. A hierarchical genetic algorithm and mixed-integer linear programming-based stochastic optimization of the configuration of integrated trigeneration energy systems. *Clean Technol. Environ. Policy* **2021**, *23*, 1913–1927. [CrossRef]
2. Cao, Y.; Wang, Q.; Wang, Z.; Jermsittiparsert, K.; Shafiee, M. A new optimized configuration for capacity and operation improvement of CCHP system based on developed owl search algorithm. *Energy Rep.* **2020**, *6*, 315–324. [CrossRef]
3. Hassan, R.; Das, B.; Al-Abdeli, Y. Investigation of a hybrid renewable-based grid-independent electricity-heat nexus: Impacts of recovery and thermally storing waste heat and electricity. *Energy Convers. Manag.* **2022**, *252*, 115073. [CrossRef]
4. Peng, D.; Wu, H.; Wang, L. Comprehensive energy cooperative optimization model based on energy conversion efficiency considering investment benefit. *Int. J. Energy Res.* **2021**, *45*, 2997–3015. [CrossRef]
5. Li, N.; Zhao, X.; Shi, X.; Pei, Z.; Mu, H.; Taghizadeh-Hesary, F. Integrated energy systems with CCHP and hydrogen supply: A new outlet for curtailed wind power. *Appl. Energy* **2021**, *303*, 117619. [CrossRef]
6. Das, B.; Al-Abdeli, Y.; Kothapalli, G. Integrating renewables into stand-alone hybrid systems meeting electric, heating, and cooling loads: A case study. *Renew. Energy* **2021**, *180*, 1222–1236. [CrossRef]

7. Dong, H.; Fang, Z.; Ibrahim, A.; Cai, J. Optimized Operation of Integrated Energy Microgrid with Energy Storage Based on Short-Term Load Forecasting. *Electronics* **2022**, *11*, 22. [CrossRef]
8. Namnabat, M.; Nazari, M.E. The effects of PV/T utilization on short-term scheduling of integrated distributed CHP system. In Proceedings of the 24th Electrical Power Distribution Conference, Khoramabad, Iran, 29 June 2019; pp. 1–6.
9. Yan, R.; Lu, Z.; Wang, J.; Chen, H.; Wang, J.; Yang, Y.; Huang, D. Stochastic multi-scenario optimization for a hybrid combined cooling, heating and power system considering multi-criteria. *Energy Convers. Manag.* **2021**, *233*, 113911. [CrossRef]
10. Li, L.; Liu, Z.; Tseng, M.; Zheng, S.; Lim, M. Improved tunicate swarm algorithm: Solving the dynamic economic emission dispatch problems. *Appl. Soft Comput.* **2021**, *108*, 107504. [CrossRef]
11. Zhang, Y.; Yan, Z.; Zhou, C.; Wu, T.; Wu, T.; Wang, Y. Capacity allocation of HESS in micro-grid based on ABC algorithm. *Int. J. Low-Carbon Technol.* **2020**, *15*, 496–505. [CrossRef]
12. Fragiacomo, P.; Lucarelli, G.; Genovese, M.; Florio, G. Multi-objective optimization model for fuel cell-based poly-generation energy systems. *Energy* **2021**, *237*, 121823. [CrossRef]
13. Gu, W.; Tang, Y.; Peng, S.; Wang, D.; Sheng, W.; Liu, K. Optimal configuration and analysis of combined cooling, heating, and power microgrid with thermal storage tank under uncertainty. *J. Renew. Sustain. Energy* **2015**, *7*, 2125–2141. [CrossRef]
14. Olamaei, J.; Nazari, M.; Bahravar, S. Economic environmental unit commitment for integrated CCHP-thermal-heat only system with considerations for valve-point effect based on a heuristic optimization algorithm. *Energy* **2018**, *159*, 737–750. [CrossRef]
15. Kuang, J.; Zhang, C.; Bo, S. Stochastic dynamic solution for off-design operation optimization of combined cooling, heating, and power systems with energy storage. *Appl. Therm. Eng.* **2019**, *163*, 114356. [CrossRef]
16. Soheyli, S.; Mehrjoo, M.; Mayam, M. Modeling and optimal resources allocation of a novel tri-distributed generation system based on sustainable energy resources. *Energy Convers. Manag.* **2017**, *143*, 1–22. [CrossRef]
17. Li, F.; Sun, B.; Zhang, C.; Liu, C. A hybrid optimization-based scheduling strategy for combined cooling, heating, and power system with thermal energy storage. *Energy* **2019**, *188*, 115948. [CrossRef]
18. Faridnia, N.; Habibi, D.; Lachowicz, S.; Kavousifard, A. Optimal scheduling in a microgrid with a tidal generation. *Energy* **2019**, *171*, 435–443. [CrossRef]
19. Sun, X.; Wang, G.; Xu, L.; Yuan, H.; Youse, N. Optimal performance of a combined heat-power system with a proton exchange membrane fuel cell using a developed marine predators algorithm. *J. Clean. Prod.* **2021**, *284*, 124776. [CrossRef]
20. Mao, Y.; Wu, J.; Zhang, W. An Effective Operation Strategy for CCHP System Integrated with Photovoltaic/Thermal Panels and Thermal Energy Storage. *Energies* **2020**, *13*, 6418. [CrossRef]
21. Chen, K.; Pan, M. Operation optimization of combined cooling, heating, and power superstructure system for satisfying demand fluctuation. *Energy* **2021**, *237*, 121599. [CrossRef]
22. Zhao, H.; Wang, X.; Wang, Y.; Li, B.; Lu, H. A dynamic decision-making method for energy transaction price of CCHP microgrids considering multiple uncertainties. *Int. J. Electr. Power Energy Syst.* **2021**, *127*, 106592. [CrossRef]
23. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.; Al-qaness, M.; Gandomi, A. Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
24. Wang, S.; Jia, H.; Abualigah, L.; Liu, Q.; Zheng, R. An Improved Hybrid Aquila Optimizer and Harris Hawks Algorithm for Solving Industrial Engineering Optimization Problems. *Processes* **2021**, *9*, 1551. [CrossRef]
25. Li, L.; Liu, Z.; Tseng, M.; Jantarakolica, K.; Lim, M. Using enhanced crow search algorithm optimization-extreme learning machine model to forecast short-term wind power. *Expert Syst. Appl.* **2021**, *184*, 115579. [CrossRef]
26. Ma, B.; Lu, P.; Liu, Y.; Zhou, Q.; Hu, Y. Shared seagull optimization algorithm with mutation operators for global optimization. *AIP Adv.* **2021**, *11*, 125217. [CrossRef]
27. Cho, H.; Smith, A.; Mago, P. Combined cooling, heating and power: A review of performance improvement and optimization. *Appl. Energy* **2014**, *136*, 168–185. [CrossRef]
28. Liu, Z.; Li, L.; Liu, Y.; Liu, J.; Li, H.; Shen, Q. Dynamic economic emission dispatch considering renewable energy generation: A novel multi-objective optimization approach. *Energy* **2021**, *235*, 121407. [CrossRef]
29. Nami, H.; Anvari-Moghaddam, A.; Nemati, A. Modeling and analysis of a solar boosted biomass-driven combined cooling, heating and power plant for domestic applications. *Sustain. Energy Technol. Assess.* **2021**, *47*, 101326. [CrossRef]
30. Wei, D.; Ji, J.; Fang, J.; Yousefi, N. Evaluation and optimization of PEM Fuel Cell-based CCHP system based on Modified Mayfly Optimization Algorithm. *Energy Rep.* **2021**, *7*, 7663–7674. [CrossRef]
31. Tonekaboni, N.; Feizbahr, M.; Tonekaboni, N.; Jiang, G.; Chen, H. Optimization of Solar CCHP Systems with Collector Enhanced by Porous Media and Nanofluid. *Math. Probl. Eng.* **2021**, *2021*, 9984840. [CrossRef]
32. Li, H.; Kang, S.; Lu, L.; Liu, L.; Zhang, X.; Zhang, G. Optimal design and analysis of a new CHP-HP integrated system. *Energy Convers. Manag.* **2017**, *146*, 217–227. [CrossRef]
33. Nazari, M.E.; Bahravar, S.; Olamaei, J. Effect of storage options on price-based scheduling for a hybrid trigeneration system. *Int. J. Energy Res.* **2020**, *44*, 7342–7356. [CrossRef]
34. Li, G.; Wang, R.; Zhang, T.; Ming, M. Multi-Objective Optimal Design of Renewable Energy Integrated CCHP System Using PICEA-g. *Energies* **2018**, *11*, 743. [CrossRef]
35. Leonzio, G. An innovative trigeneration system using biogas as renewable energy. *Chin. J. Chem. Eng.* **2018**, *26*, 1179–1191. [CrossRef]
36. Dong, X.; Quan, C.; Jiang, T. Optimal Planning of Integrated Energy Systems Based on Coupled CCHP. *Energies* **2018**, *11*, 2621. [CrossRef]

37. Zhang, J.; Cho, H.; Mago, P.; Zhang, H.; Yang, F. Multi-Objective Particle Swarm Optimization (MOPSO) for a Distributed Energy System Integrated with Energy Storage. *J. Therm. Sci.* **2019**, *28*, 1221–1235. [CrossRef]
38. Mohsenipour, M.; Ebadollahi, M.; Rostamzadeh, H.; Amidpour, M. Design and evaluation of a solar-based trigeneration system for a nearly zero energy greenhouse in arid region. *J. Clean. Prod.* **2020**, *254*, 119990. [CrossRef]
39. De Souza, R.; Casisi, M.; Micheli, D.; Reini, M. A Review of Small-Medium Combined Heat and Power (CHP) Technologies and Their Role within the 100% Renewable Energy Systems Scenario. *Energies* **2021**, *14*, 5338. [CrossRef]
40. Zhao, H.; Lu, H.; Wang, X.; Li, B.; Wang, Y.; Liu, P.; Ma, Z. Research on Comprehensive Value of Electrical Energy Storage in CCHP Microgrid with Renewable Energy Based on Robust Optimization. *Energies* **2020**, *13*, 6526. [CrossRef]
41. Gao, L.; Hwang, Y.; Cao, T. An overview of optimization technologies applied in combined cooling, heating and power systems. *Renew. Sustain. Energy Rev.* **2019**, *114*, 109344. [CrossRef]
42. Wen, Q.; Liu, G.; Wu, W.; Liao, S. Genetic algorithm-based operation strategy optimization and multi-criteria evaluation of distributed energy system for commercial buildings. *Energy Convers. Manag.* **2020**, *226*, 113529. [CrossRef]
43. Yousif, M.; Ai, Q.; Gao, Y.; Wattoo, W.; Jiang, Z.; Hao, R. Application of Particle Swarm Optimization to a Scheduling Strategy for Microgrids Coupled with Natural Gas Networks. *Energies* **2018**, *11*, 3499. [CrossRef]
44. Wang, F.; Zhou, L.; Wang, B.; Wang, Z.; Shafie-Khah, M.; Catalao, J. Modified Chaos Particle Swarm Optimization-Based Optimized Operation Model for Stand-Alone CCHP Microgrid. *Appl. Sci.* **2017**, *7*, 754. [CrossRef]
45. Guo, Z.; Yang, B.; Han, Y.; He, T.; He, P.; Meng, X.; He, X. Optimal PID Tuning of PLL for PV Inverter Based on Aquila Optimizer. *Front. Energy Res.* **2022**, *9*, 812467. [CrossRef]
46. Fatani, A.; Dahou, A.; Al-qaness, M.; Lu, S.; Elaziz, M. Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System. *Sensors* **2022**, *22*, 140. [CrossRef] [PubMed]
47. Deru, M.; Field, K.; Studer, D. U.S. Department of Energy Commercial Reference Building Models of the National Building Stock. 2011. Available online: https://www.energy.gov/eere/buildings/commercial-reference-buildings (accessed on 9 September 2020).
48. Wilson, E. Commercial and Residential Hourly Load Profiles for all TMY3 Locations in the United States. 2020. Available online: https://openei.org/doe-opendata/dataset/commercial-and-residential-hourly-load-profiles-for-all-tmy3-locations-in-the-united-states (accessed on 9 September 2020).

symmetry

Article

IoT Botnet Detection Using Salp Swarm and Ant Lion Hybrid Optimization Model

Ruba Abu Khurma [1], Iman Almomani [1,2] and Ibrahim Aljarah [1,*]

[1] King Abdullah II School of Information Technology, The University of Jordan, Amman 11942, Jordan; ruba_abukhurma@yahoo.com or RBA9150539@fgs.ju.edu.jo (R.A.K.); imomani@psu.edu.sa (I.A.)
[2] Security Engineering Lab, Computer Science Department, Prince Sultan University, Riyadh 11586, Saudi Arabia
* Corresponding: i.aljarah@ju.edu.jo

Abstract: In the last decade, the devices and appliances utilizing the Internet of Things (IoT) have expanded tremendously, which has led to revolutionary developments in the network industry. Smart homes and cities, wearable devices, traffic monitoring, health systems, and energy savings are typical IoT applications. The diversity in IoT standards, protocols, and computational resources makes them vulnerable to security attackers. Botnets are challenging security threats in IoT devices that cause severe Distributed Denial of Service (DDoS) attacks. Intrusion detection systems (IDS) are necessary for safeguarding Internet-connected frameworks and enhancing insufficient traditional security countermeasures, including authentication and encryption techniques. This paper proposes a wrapper feature selection model (SSA–ALO) by hybridizing the salp swarm algorithm (SSA) and ant lion optimization (ALO). The new model can be integrated with IDS components to handle the high-dimensional space problem and detect IoT attacks with superior efficiency. The experiments were performed using the N-BaIoT benchmark dataset, which was downloaded from the UCI repository. This dataset consists of nine datasets that represent real IoT traffic. The experimental results reveal the outperformance of SSA–ALO compared to existing related approaches using the following evaluation measures: TPR (true positive rate), FPR (false positive rate), G-mean, processing time, and convergence curves. Therefore, the proposed SSA–ALO model can serve IoT applications by detecting intrusions with high true positive rates that reach 99.9% and with a minimal delay even in imbalanced intrusion families.

Keywords: Internet of Things; IoT; botnets; attack detection; feature selection; ant lion optimization; security; DoS; malware; salp swarm optimization

Citation: Abu Khurma, R.; Almomani, I.; Aljarah, I. IoT Botnet Detection Using Salp Swarm and Ant Lion Hybrid Optimization Model. *Symmetry* **2021**, *13*, 1377. https://doi.org/10.3390/sym13081377

Academic Editor: Ming-Chin Chuang

Received: 7 July 2021
Accepted: 23 July 2021
Published: 28 July 2021

1. Introduction

A set of connected devices that use a wireless connection to communicate, sense, compute, process, share, and store information over the Internet defines the new technology trend Internet of Things (IoT). IoT is a set of connected devices, including electronic, physical objects, and embedded objects, that communicate through the Internet without human intervention (machine-to-machine). Recently, there has been a massive proliferation in the number of connected devices in IoT. The expected number of connected objects in 2021 is 13.8 billion, and it is expected to jump to 30.9 billion by 2025 (https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/ (accessed on 8 March 2021)). IoT applications are strongly connected to humans' daily lives, including the areas of health, agriculture, fleet management, hospitality, and many others.

IoT is characterized by low computational memory, battery, streaming bandwidth, and processing unit. Due to these characteristics, IoT becomes more susceptible to security breaches. The plug-and-play facility of IoT devices and the original passwords from their manufacturers make them more attractive to brute-force and botnet attacks. Other issues that increase the vulnerability of IoT to software and hardware security threats are the heterogeneity and scalability with other interconnected networks.

IoT devices attacked by malicious software are compromised by a botnet. This botnet is created by brute-forcing techniques or weak credential exploitation to compromise the victim device [1]. Once the device is compromised, the attacker gains control by downloading malicious binaries to enrol into the IoT botnet [2,3].

In general, three main layers encompass IoT: perception, transportation, and the application layers. Each layer applies different standards, making it easier to be attacked by different security breaches. Specifically, the transportation layer includes several technologies for communication, so it can be attacked by denial-of-service (DoS) or distributed denial-of-service (DDoS) attacks. Five years ago, a remarkable DDoS attack occurred involving the Dyn (https://www.kaspersky.com/blog/attack-on-dyn-explained/13325/ (accessed on 26 October 2016)) domain name service provider. This attack launched the Mirai botnet to flood the servers, disrupt many functionalities, and stop the services of many websites, including Twitter, CNN, PayPal, and Netflix. Intrusions threaten mainly the system's confidentiality, integrity, availability, and authenticity. Traditional security countermeasures, including authentication protocols and encryption techniques, might not be sufficient to provide acceptable security levels for the highly scalable and interconnected IoT. Many emergent protection technologies work against these intrusions, such as blockchain, fog, and cloud computing technology. However, these technologies still have shortcomings related to time latency and scalability issues [4]. Intrusion detection systems (IDSs) are essential and crucial for IoT. In this solution, hardware and software are used to monitor the network and discover malicious behaviors. Typically, there are different types of IDSs, which could be based on statistics, machine learning, or others [5–11].

IDSs consist of three primary components: the sensing component to gather the information from the environment and the analysis and reporting components. In the analysis components, different intelligent data mining techniques help to process the massive volume of monitored data and capture the abnormal and malicious patterns. Therefore, the analysis component is the smart component in IDS that deploys smart and lightweight security models to protect the network.

IoT involves many connected devices with a high amount of collected high-dimensional data. Such colossal data need data mining techniques to process them, including feature selection (FS) [12].

FS is a data mining technique used to distinguish irrelevant and symmetrical features that may reduce the classifier's performance. FS contributes to reducing the dimensionality, enhancing the classification performance, and even reducing the training time. Using traditional search techniques, such as exhaustive search, yields exponential running time. Thus, if a dataset has N features, the size of the entire feature space is 2^N. This is practically impossible, especially with a medium and large number of features. Different search methods have been investigated in recent years, but most of them have suffered from the local minima. Recently, metaheuristic algorithms have been applied widely and efficiently for the optimization of the FS problem and have achieved promising results [13].

This work proposes a hybrid model using a salp swarm algorithm (SSA) and ant lion optimization (ALO). The new model is called SSA–ALO, which integrates the power points of both algorithms into one method. The hybridization exploits the ability to search globally (exploration) of the ALO and the ability to search locally by the SSA, consequently achieving a balance between global search and local search of feature space and increasing the opportunity to reach the optimal solution (best asymmetrical features subset) and alleviate the local minima problem. The SSA algorithm has one parameter that adaptively decreases across the iterations of the optimization process. Thus, the algorithm explores several regions at the beginning of the optimization process and focuses on promising regions later on. As a bonus, follower salps update their positions gradually according to other salps in the swarm, which prevents the optimizer from falling into local minima. The SSA maintains the best-found individual so that it is reserved even if agents become weakened. In the SSA, the leader salp moves based on the position of the food source only, which is the best salp found so far, so the leader always is capable of searching globally and locally around the food source in the search space.

ALO has two types of individuals: ants and ant lions. Ant lions are the best solutions found so far. Their positions are replaced whenever a fitter ant is found. Ants are moving around in the search space continuously. The positions of ants are changed based on the positions of ant lions. The position update strategy of ants is based on selecting an ant lion using a roulette wheel in combination with the best solution. Thus, a given ant updates its position based on these two agents. The selection of an agent randomly using a roulette wheel encourages diversification in the search space. Therefore, ants can move randomly in the search space and explore more regions without stagnating in local minima. The avoidance of local minima is a significant merit of ALO, which gives it superiority against other methods such as PSO. Moreover, it has few parameters compared to PSO and GA. In SSA, the swarm leader is chosen to be the agent with the highest fitness value. Hence, the agents of low fitness values have no chance of leading the swarm. This decreases the exploration capability of the algorithm and supports its exploitative power. In contrast, ALO keeps track of all agents in the swarm and uses a roulette wheel together with the current best agent to lead the swarm. This indicates that low-fitness agents can participate in guiding other agents in the swarm besides the best-found agent. This supports the exploratory behavior of the ALO. Integrating the principles of ALO and SSA into one algorithm can support the exploration/exploitation trade-off. The proposed hybrid algorithm keeps the ants and ant lion swarms in motion. However, it uses the ideas of leadership assignments from both ALO and SSA to provide more trade-offs between global search and local search. The proposed SSA–ALO algorithm takes the merits of ALO by updating low-fitness agents (ants) using ALO principles. On the other hand, it uses the merits of SSA by updating the high-fitness agents (ant lions) using SSA principles that have to maintain faster convergence.

The remaining parts of the paper are as follows: Section 2 introduces a related studies review. Sections 3 and 4 present details regarding the proposed techniques. Section 5 analyzes the results. Finally, Section 7 provides a paper summary and suggests the future works.

2. Related Works

In the literature, researchers have been attracted by new technologies of data mining. Therefore, they have utilized them for security breaches and botnet attacks in IoT in an efficient way. In [14], neural network (NN) and the negative selection algorithm were used to predict intrusions in IoT. However, in this study, the performance of the IDs was limited. Mehmood [15], used a multi-agent system to detect DDoS attacks. Another study [16] classified the machine learning methods used for security in IoT. In [17], the authors used and compared a set of classifiers to determine their performance in detecting malicious activities. The results showed that NB achieved the worst performance among RF and GB. Principal component analysis (PCA) and fuzzy clustering were used to detect intrusions in IoT. The yielded efficiency of the proposed IDS was promising. However, the system suffered from the scalability problem, so it became inefficient when the amount of data increased [18].

Rathore, in [19], proposed an extreme learning machine (ELM) for attack detection in IoT. The new method outperformed other traditional machine learning methods in terms of accuracy. Moustafa [20] proposed an ensemble learning method for IoT security detection. The proposed method was used to detect mainly three kinds of attacks in IoT.

Several nature-inspired algorithms were used to deal with security issues by developing IDS for IoT. Hamamoto [21] proposed IDS based on a genetic algorithm. In [22], a genetic algorithm was used to detect insider threats. The authors in [23] developed an IDS using multilayer perception and an artificial bee colony algorithm to detect malicious patterns. Ali in [24] proposed IDS based on particle swarm optimization (PSO) to detect attacks. In [25], Bayesian networks and C4.5 were used with the firefly algorithm to perform FS and classify the data in the network. PSO, GA, and ant colony algorithms were proposed as a layered model in [26] along with five rule-based classifiers.

In all mentioned studies, the proposed IDSs were used to handle the data traffic in the traditional networks or typical wireless sensor networks (WSNs). This means that they were not specifically developed for IoT networks. Hence, they may be insufficient for the evolved IoT networks [27]. In the literature, there are few studies on the use of metaheuristic algorithms for detecting intrusions in IoT networks and using metaheuristic FS to enhance the detection of attacks.

In [28], RF and the bat algorithm (BA) were integrated to perform FS for IoT security. The achieved performance results were superior to those of other used methods such as SVM, AdaBoost, and decision tree (DT). Xue [29] applied a differential evolution algorithm. In [30], Popoola proposed a wrapper FS to detect intrusions in networks by using a differential evolution algorithm. Guendouzi developed an FS method to detect intrusions using the biogeography-based optimization (BBO) algorithm [31]. In [32], the author proposed a genetic IDS. There is no existing solution in the literature that uses the integration of SSA and ALO for FS in IDSs, especially to detect botnets in IoT networks. In [8], the authors proposed a deep multi-layer classification intrusion detection approach. They used two stages for detecting an intrusion and the type of intrusion. They also applied an oversampling technique to enhance the classification results. The experiments showed that the best settings of the proposed approach included oversampling by the intrusion type identification label (ITI), 150 neurons for the Single-Hidden Layer Feed-Forward Neural Network (SLFN), and 2 layers and 150 neurons for LSTM. The results showed that the proposed technique outperformed the other well-known techniques in terms of the G-mean having a value of 78% compared to 75% for KNN and less than 50% for the other techniques.

In [6], the authors proposed a new approach for intrusion detection. They integrated the unsupervised (clustering), supervised (classification) and oversampling methods for carrying the task. The used classifier in the proposed approach is the Single Hidden Layer Feed-Forward Neural Network (SLFN). The oversampling method was applied to generate balanced training data. The results showed that the SLFN with the Support Vector Machine and Synthetic Minority Oversampling (SVM-SMOTE), with a ratio of 0.9 and a k-means++ clustering with k value of 3 obtained better results than other values and other classification methods.

3. Preliminaries

3.1. Salp Swarm Optimization

The salp swarm algorithm (SSA) is a swarm intelligence algorithm that was developed in [33]. It inspiration is from animals called salps that live in the sea as a chain. The chain starts with a leader salp that leads other salps. Other salps are followers of the leader salp. Figure 1 shows the swarm of salps and the individual salp.

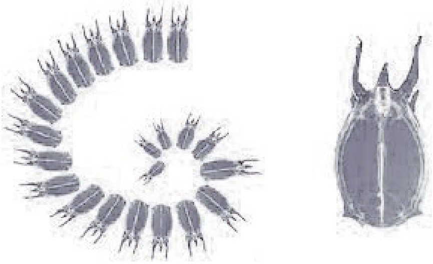


Figure 1. The chain of salps and the single salp.

The population X of n salps can be represented using a 2D matrix as shown in Equation (1). F is the food source. The matrix represents the population of salps. Each line in this matrix represents a single salp (solution). This solution or salp consists of a number of elements. The length of the salp represents the dimension of the problem (d). Actually, this is the number of dimensions or features in a dataset. The number of salps (n) represents the size of the population. Therefore, the first line in the matrix represents the first salp in the swarm, which is the leader salp, and the last line in the matrix is the follower salp number (n) in the swarm. Equation (2) explains the position relation of the leader salp (which is the first solution in the swarm of salp) and the food source.

$$X_i = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_d^1 \\ x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^n & x_2^n & \cdots & x_d^n \end{bmatrix} \tag{1}$$

The position of the leader salp is formulated as in Equation (2)

$$X_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j)\,, & c_3 \geqslant 0.5 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j)\,, & c_3 < 0.5 \end{cases} \tag{2}$$

where x_j^1 and F_j are the positions of leaders and food source in the jth dimension, respectively. c_1 is a gradually decreasing parameter across iterations, and calculated as in Equation (3), where l and L are the current iteration and the maximum iterations, respectively. The other c_2 and c_3 parameters in Equation (2) are randomly selected from [0, 1]. The parameters c_2 and c_3 are very important to direct the next position in the jth dimension towards +∞ or −∞ and determine the step size. The ub_j and lb_j are the bounds of the jth dimension.

$$C_1 = 2e^{-(\frac{4l}{L})^2} \tag{3}$$

$$X_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \tag{4}$$

In Equation (4), $i \geqslant 2$ and x_j^i represents the position of the ith follower at the jth dimension. SSA is presented in Algorithm 1.

Algorithm 1 SSA algorithm pseudo-code.

```
Input: $n$ is the # salps, $d$ is the # dimensions)
Output:Near optimal solution (F)
Initialization step x_i (i = 1, 2, . . . , n) considering  ub and lb
  while (maximum iterations is not reached) do
    Computer the fitness value of each individual
    Define F as the best individual
    Update $c_1$ by Equation (3)
    for (each individual $x_i$) do
    if $l$ == 1 then
      Change the position of the leader individual by Equation (2)
    else
      Change the position of the follower individual by Equation (4)
    end if
    Change the positions of the individuals based on the bounds of variables
  end while
  return  F
```

3.2. Ant Lion Optimization

Ant lion optimization (ALO) was developed in 2015 [34]. ALO mimics the ant lions' strategy for hunting in nature. Ant lions begin to hunt when they are larvae, while they reproduce when they become adults. Ant lion larvae build a hole in the land. This hole is the trap for insects such as ants. Ant lion larvae hide in the hole and wait for the ants. When the ant lion recognizes ant, it throws the sand out of the hole, in an attempt to catch it. Thus, the ant is captured and is unable to get out from the hole. After the ant lion consumes the prey, it tries to fix the trap for the next prey.

In the ALO algorithm, the ants move within the search space in a random way. Ants are affected by the traps that are built by ant lions. The trap size depends on the fitness value of the ant lion so that a large hole indicates the high fitness of the ant lion. Hence, the ant lion with a higher fitness value has a greater chance of catching prey. In each iteration, prey can be hunted by an ant lion. In ALO, the random walks of ants decrease in adaptive way across iterations to reflect the situation of sliding ants in the direction of the ant lion. If the ant is hunted by the ant lion and pulled towards the bottom of the trap, this is described in terms of the ant having higher fitness than the ant lion. The ant lion updates its position to the last hunted prey and builds another hole in preparation for the next hunt. Figure 2 shows the ants and the ant lion in nature. ALO is formulated in Algorithm 2.

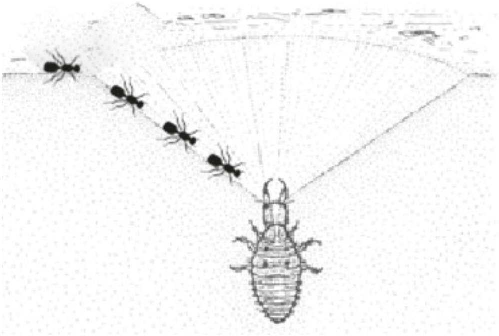


Figure 2. The ants and the ant lion.

Algorithm 2 ALO algorithm pseudo-code.

```
Input: Search space, fitness function, # ants and ant lions, # iterations ($Max_{Iter}$), $a_i$,$b_i$.
Output: The best ant lion and its fitness.
1. Initialize a random $n$ ant positions and $n$ ant lion positions.
2. Compute the fitness of all ants and ant lions.
3. Find the best ant lion
4.
  while $t \leq Max_{Iter}$ do
    for all $ant_i$ do
      (i) Use a roulette wheel to select an ant lion (building trap); as in Equations (9) and (10).
      (ii) Slide ants towards the ant lion; as in Equations (11) and (12).
      (iii) Create a random walk for anti and normalize it; as in Equation (14).
    end for
    - Compute the fitness of all ants.
    - Replace an ant lion with a fitter ant.
    - Update elite if an ant lion becomes fitter than the elite.
  end while
  5. Select the optimal ant lion position.
```

- t: the current iteration
- T: the maximum iterations
- $Antlion_j^t$: the position of ant lion j at iteration t
- Ant_i^t: the position ant i at iteration t

- c^t: is the minimum of all variables at tth iteration.
- d^t: indicates the vector including the maximum of all variables at tth iteration.
- w: a constant defined based on the current iteration ($w = 2$ when $t > 0.1T$, $w = 3$ when $t > 0.5T$, $w = 4$ when $t > 0.75T$, $w = 5$ when $t > 0.9T$, and $w = 6$ when $t > 0.95T$). The constant w can adjust the accuracy level of exploitation.
- I: I is a ratio defined based on w using the equation

$$I = 10^w \times \frac{t}{T} \tag{5}$$

- a_i: minimum random walk of $i$thvariable.
- b_i: maximum random walk of $i$thvariable.

Random walks of ants are all based on Equation (6):

$$\begin{aligned} X(t) = [0, cumsum(2r(t_1) - 1); cumsum(2r(t_2) - 1); \\ \ldots; cumsum(2r(t_T) - 1)], \end{aligned} \tag{6}$$

where cumsum is the cumulative sum, T is the maximum # iterations, t shows the step of random walk, and $r(t)$ is a function in Equation (7)

$$r(t) = \begin{cases} 1, & \text{if } rand > 0.5 \\ 0, & \text{if } rand \leq 0.5 \end{cases} \tag{7}$$

with *rand* being a random number in [0; 1].

To keep the random walks inside the search space, the minmax normalization is applied:

$$X_i^{t+1} = \frac{(X_i^t - a_i) \times (d_i - c_i^t)}{(b_i^t - a_i)} + c_i \tag{8}$$

Trapping of ants in the ant lion's hole is expressed in Equations (9) and (10)

$$c_i^t = c^t + Antlion_j^t \tag{9}$$

$$d_i^t = d_t + Antlion_j^t \tag{10}$$

Ant lions release sand outside the hole as soon as an ant is in the trap. This behavior is mathematically modeled in Equations (11) and (12)

$$c^t = \frac{c^t}{I} \tag{11}$$

$$d^t = \frac{d^t}{I} \tag{12}$$

The last step is to catch the prey and rebuild the hole. It is assumed that the hunting of ants occurs when ants have higher fitness values than their corresponding ant lions. An ant lion is then expected to update its position regarding the latest position of the hunted ant to improve its chance of catching new prey, modeled by Equation (13)

$$Antlion_j^t = Ant_i^t \ If \ f(Ant_i^t) \ is \ better \ than \ f(antlion_j^t) \tag{13}$$

Elitism is used to reserve the best solution(s) across iterations. The random walk of an ant is led by the selected ant lion and the elite ant lion and, therefore, the new position of an ant is formulated as in Equation (14)

$$Ant_i^t = \frac{R_R^t + R_E^t}{2} \tag{14}$$

where R_A^t is the random walk around the ant lion selected by the roulette wheel, and R_E^t is the random walk around the elite ant lion.

4. The SSA–ALO Hybrid Model for Feature Selection

The SSA algorithm has many promising features that make it favorable for global optimization, particularly for FS problems. In general, SSA has a simple structure, is efficient, and has an implementable methodology. Furthermore, SSA has one parameter for balancing the global search and local search. It adaptively decreases across the iterations of the optimization process. Thus, the algorithm explores several regions at the beginning of the optimization process and focuses on promising regions later on. As a bonus, follower salps update their positions gradually according to other salps in the swarm, which prevents the optimizer from falling into local minima. The SSA maintains the best-found individual so that it is reserved even if agents become weakened. In the SSA, the leader salp moves according to the food source position. This is the best salp found so far. Therefore, the leader always is able to search globally and locally around the food source in the search space.

ALO has two types of individuals: ants and ant lions. Ant lions are the best solutions found so far. Their positions are replaced whenever a fitter ant is found. Ants are moving around in the search space continuously. The positions of ants are changed based on the positions of ant lions. The position update strategy of ants is based on selecting an ant lion using a roulette wheel in combination with the best solution. Thus, a given ant updates its position based on these two agents. The selection of an agent randomly using a roulette wheel encourages diversification in the search space. Therefore, ants can move randomly in the search space and explore more regions without stagnating in local minima. The avoidance of local minima is a significant merit of ALO, which gives it superiority against other methods such as PSO. Moreover, it has few parameters compared to PSO and GA.

The exploration/exploitation capability of a particular algorithm depends directly on how the swarm leader is announced. In SSA, the swarm leader is chosen to be the agent with the highest fitness value. Hence, the agents of low fitness values have no chance of leading the swarm. This decreases the exploration capability of the algorithm and supports its exploitative power. In contrast, ALO keeps track of all agents in the swarm and uses a roulette wheel together with the current best agent to lead the swarm. This indicates that low-fitness agents can participate in guiding other agents in the swarm besides the best-found agent. This supports the exploratory behavior of the ALO. Integrating the principles of ALO and SSA into one algorithm can support the exploration/exploitation trade-off. The proposed hybrid algorithm keeps the ants' and ant lions' swarms in motion. However, it uses the ideas of leadership assignments from both ALO and SSA to provide more trade-offs between global search and local search. The proposed SSA–ALO algorithm takes the merits of ALO by updating low-fitness agents (ants) using ALO principles. On the other hand, it uses the merits of SSA by updating the high-fitness agents (ant lions) using SSA principles that have to maintain faster convergence. The SSA–ALO is formally given in Algorithm 3. The overall botnet detection system is represented by the flowchart in Figure 3.

The proposed SSA–ALO algorithm has sufficient exploration capabilities because:

- It applies a roulette wheel as a selection mechanism of individuals. This affects the swarm of ants.
- It changes the size of the random walk adaptively as in SSA. This affects the ant lions swarm.
- The size of the random walks is adaptive as in ALO. This affects the population of ants.
- All members of the population are repositioned, rather than only the ant population, as in ALO.

Algorithm 3 The proposed hybrid SSA–ALO algorithm.

```
Input: Search space, fitness function, # ants and ant lions, # iterations ($Max_{Iter}$).
Output: The optimal ant lion and its fitness.
1. Initialize the random $n$ ant positions and $n$ ant lion positions.
2. Compute the fitness of all ants and ant lions.
3. Find the fittest ant lion (the elite)
4.
  while $t \leq Max_{Iter}$ do
    for all $ant_i$ do
      (i) choose an ant lion using roulette wheel (building trap).
      (ii) Slide ants towards the ant lion.
      (iii) build a random walk for $ant\_i$ and normalize it.
    end for
    - Compute the fitness of all ants.
    - Change the position of an ant lion with a fitter ant (catching a prey).
    - Select the leading salp from the ant lion population based on its fitness.
    - Update the exploration rate parameter $c_1$
    - Update ant lion positions
    - Update elite if an ant lion becomes fitter than the elite.
  end while
  5. Select the optimal ant lion position.
```

Furthermore, the SSA–ALO has exploitation power:

- It is arget-driven, which is realized from SSA, where the best solution is the leader and remaining solutions are the followers. This helps to improve the ant lions walk.
- The random steps of both SSA and ALO are reduced over time.
- The fitter ants are replaced with an ant lion as in ALO concepts.

The proposed SSA–ALO algorithm represents a wrapper-based FS framework. The main issue related to implementing the wrapper approach is using a learning (classification) algorithm. In the proposed algorithm, we have used the K-nearest neighbor (KNN) technique to determine the feature subsets' goodness. The fitness function in the wrapper approach can be formulated as in Equation (15) and will be used to assess the quality of individual i at iteration t (I_i^t). The main objective is to minimize FPR and $1 - TPR$.

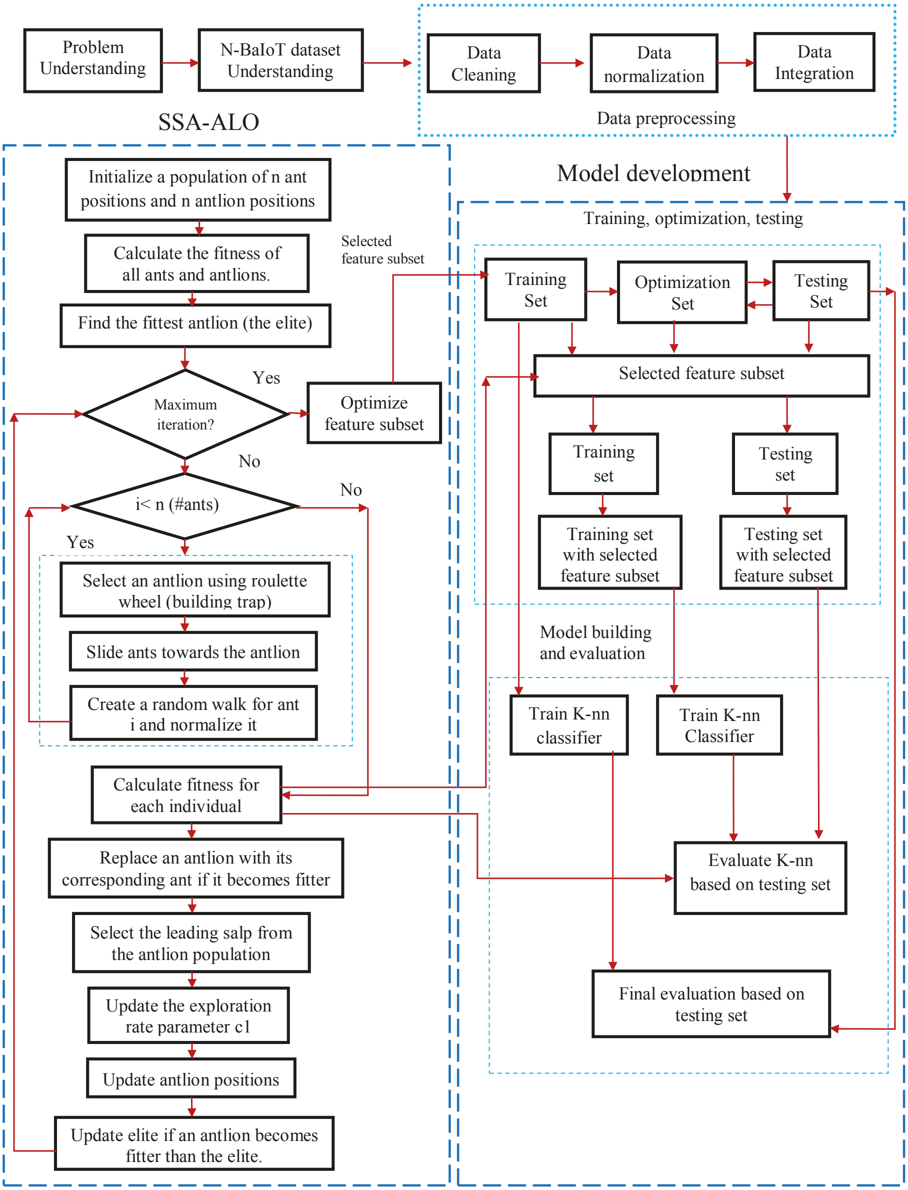
$$Fitness_i^t = FPR_i^t \times (1 - TPR_i^t) \tag{15}$$

In the wrapper framework, evaluating a single solution is costly because it requires a training–testing process using the classification algorithm. For this reason, the choice of an efficient search algorithm is necessary. In this paper, the integration of ALO and SSA is applied to perform an adaptive search in the feature search space. The main target is to increase the classification performance to the maximum and reduce the number of features to the minimum simultaneously. Iteratively, the ALO selects an ant lion randomly using a roulette wheel mechanism. Furthermore, the ants implement a random walk around the best ant lion (elite). Based on the latest two random walks, the ants adjust their positions. If the ant has higher fitness than the ant lion, the ant lion eats it and adjusts its position to the ant's position. Ant lions adjust their positions based on SSA principles, where the leader salp is selected among ant lions. This algorithm is applied iteratively and depends on the exploration rate, which decreases throughout the optimization process.

As mentioned previously, the solution is limited by two values, 0/1. Therefore, a continuous solution with real values needs to be mapped into a binary solution using Equation (16)

$$y_{ij} = \begin{cases} 0, & \text{if}(x_{ij} < 0.5) \\ 1, & \text{otherwise} \end{cases} \tag{16}$$

where x_{ij} is the continuous value of solution i in dimension j, and y_{ij} is the binary representation of solution vector x.



Problem Understanding
N-BaIoT dataset Understanding
Data Cleaning
Data normalization
Data Integration
Data preprocessing
SSA-ALO
Initialize a population of n ant positions and n antlion positions
Calculate the fitness of all ants and antlions.
Find the fittest antlion (the elite)
Maximum iteration?
Yes
Optimize feature subset
No
i< n (#ants)
No
Yes
Select an antlion using roulette wheel (building trap)
Slide ants towards the antlion
Create a random walk for ant i and normalize it
Calculate fitness for each individual
Replace an antlion with its corresponding ant if it becomes fitter
Select the leading salp from the antlion population
Update the exploration rate parameter c1
Update antlion positions
Update elite if an antlion becomes fitter than the elite.
Selected feature subset
Model development
Training, optimization, testing
Training Set
Optimization Set
Testing Set
Selected feature subset
Training set
Testing set
Training set with selected feature subset
Testing set with selected feature subset
Model building and evaluation
Train K-nn classifier
Train K-nn Classifier
Evaluate K-nn based on testing set
Final evaluation based on testing set

Figure 3. The proposed methodology diagram.

5. Experiments, Results, and Discussion

N-BaIoT dataset [35] is a benchmark dataset that was downloaded from the machine learning repository [36] at UCI (the University of California Irvine). This dataset represents IoT traffic generated after implementing two common botnet families called BASHLITE and Mirai, in addition to the implementation of the network's normal behavior.

A test bed was built to run BASHLITE and Mirai botnets consisting of nine devices connected wirelessly via Wi-Fi. Their access point was connected wired-based with the attacker side (mainly C&C server). The types of devices included were two doorbells, a thermostat, a baby monitor, four security cameras, and a webcam.

The main attack executed by these botnets was the DoS attack. DoS attacks or distributed DoS attacks target mainly the availability security requirement of the IoT applications. DoS takes advantage of the resource constraint of the IoT devices, including energy, memory, processing capabilities, and bandwidth, to initiate or inject attacks that drain the devices and network resources and make their services unavailable to the concerned users. Table 1 shows the types of attacks that both botnet families carry out. Both BASHLITE and Mirai botnets execute an auto-scanning phase to scan the IoT and discover the vulnerable devices to inject them with malicious code (malware). The attacker's C&C (command and control) servers can control this malware remotely. The main purpose of this phase is to propagate the malware and enlarge the botnet.

Table 1. N-BaIoT dataset main attack categories and record distribution.

BASHLITE					**Mirai**		
Scanning	Spamming		Flooding		Scanning	Flooding	
Auto Scan 255,111	Junk 261,789	COMBO 515,156	TCP 859,850	UDP 946,366	Auto Scan 537,979	TCP 1,377,120	UDP 1,753,303
255,111	776,945		1,806,216		537,979	3,130,423	

Moreover, both the TCP and UDP transport layers' protocols were utilized to flood the IoT by sending too many fake packets too fast in order to deplete the device and network resources. Since TCP is a connection-oriented protocol, specific TCP packets were created and transmitted, including synchronization (SYN) and acknowledgement (ACK) packets.

Additionally, the BASHLITE botnet implemented spamming attacks, whether through sending spam data (junk) or opening fake connections with specific ports and IP addresses (COMBO) to book them just for DoS purposes. The total records per attack category are shown in Table 1.

Table 2 presents the 23 recorded features. Table 3 presents the number of normal and abnormal packets for each IoT device in the N-BaIoT Dataset. Table 4 presents a new N-BaIoT dataset that resulted from combining the data based on object type and the numbers of packets that were utilized in different stages. The new version of the N-BaIoT dataset is called the NN-BaIoT dataset. This is the one used for evaluating the proposed method.

Table 2. The N-BaIoT dataset features.

Value	Measure	Aggregated by	# Features
The size of outbound packets	Mean Variance	Source IP, Source MAC-IP, Channel, Socket	8
Packet count	Number	Source IP, Source MAC-IP, Channel, Socket	4
Packet jitter	Mean, Variance, Number	Channel	3
Packet size	Magnitude, Radius, Covariance, Correlation Coefficient	Channel Socket	8

Table 3. N-BaIoT dataset records according to IoT device and attack class.

Model of Device	Type of Device	Benign	BASHLITE					Mirai					Tot Attacks
			Combo	Junk	Scan	TCP	UDP	Ack	Scan	Syn	UDP	UDPPlain	
Danmini	Doorbell	40,395	59,718	29,068	29,849	92,141	105,874	102,195	107,685	122,573	237,665	81,982	968,750
Ennio	Doorbell	34,692	53,014	29,797	28,120	101,536	103,933	0	0	0	0	0	316,400
Ecobee	Thermostat	13,111	53,012	30,312	27,494	95,021	104,791	113,285	43,192	116,807	151,481	87,368	822,763
Philips B120N/10	Baby monitor	160,137	58,152	28,349	27,859	92,581	105,782	91,123	103,621	118,128	217,034	80,808	923,437
Provision PT-737E	Sec. camera	55,169	61,380	30,898	29,297	104,510	104,011	60,554	96,781	65,746	156,248	56,681	766,106
Provision PT-838	Sec. camera	91,555	57,530	29,068	28,397	89,387	104,658	57,997	97,096	61,851	158,608	53,785	738,377
SH XCS7-1002-WHT	Sec. camera	42,784	54,283	28,579	27,825	88,816	103,720	111,480	45,930	125,715	151,879	78,244	816,471
SH XCS7-1003-WHT	Sec. camera	17,936	59,398	27,413	28,572	98,075	102,980	107,187	43,674	122,479	157,084	84,436	831,298
Samsung SNH 1011 N	Webcam	46,817	58,669	28,305	27,698	97,783	110,617	0	0	0	0	0	323,072
		502,596	515,156	261,789	255,111	859,850	946,366	643,821	537,979	733,299	1,229,999	523,304	6,506,674

Table 4. The division of NN-BaIoT dataset into training, optimization, and testing stages.

Device Type	Training	Optimization		Testing	
	No. of Benign Instances	**No. of Benign Instances**	**No. of Malicious Instances**	**No. of Benign Instances**	**No. of Malicious Instances**
Baby monitor	53,379	53,379	307,812	53,379	615,625
Doorbell	25,029	25,029	428,383	25,029	856,767
Security camera	69,148	69,148	1,050,750	69,148	2,101,502
Thermostat	4371	4370	274,254	4370	548,509
Webcam	15,607	15,605	107,690	15,605	215,382

All optimization algorithms were implemented using the EvoloPy-FS framework [13]. This is a publicly available FS swarm intelligence framework coded in Python. EvoloPy-FS consists of several metaheuristic algorithms, which can be downloaded from (www.evo-ml.com) (accessed on 28 March 2021). All results were recorded based on population size and maximum number of iterations equal to 10 and 100, respectively.

The proposed SSA–ALO algorithm was assessed using three evaluation measures that are widely used for testing the capability of attack detection; in the following equations, TP indicates true positives, FN indicates false negatives, FP indicates false negatives, TN indicates true negatives, TPR indicates the true positive rate as in Equation (17), FPR indicates the false positive rate as in Equation (18), and G-mean indicates the square root of of TPR multiplied by $1 - FPR$ as in Equation (19).

$$TPR = \frac{TP}{TP + FN} \tag{17}$$

$$FPR = \frac{FP}{FP + TN} \tag{18}$$

$$G - mean = \sqrt{TPR \times (1 - FPR)} \tag{19}$$

These evaluation measures were applied to the testing part of the dataset. The best anomaly detection was achieved when larger values of $G - mean$ and TPR were obtained. Moreover, lower values of FPR were required to increase the robustness of the anomaly detection approach. Consequently, the perfect anomaly detection occurred when both $G - mean$ and TPR equaled one and FPR equaled zero. Table 5 shows the TPR and FPR values for the SSA, ALO, and the proposed SSA–ALO algorithms. In addition, we compared the results with other anomaly detection algorithms from the literature. These are IF, LOF, OCSVM, and GWO-OCSVM. All the algorithms were applied over the NN-BaIoT dataset at the testing stage. TPR for anomaly detection means predicting the attack whenever it occurs. FPR means predicting benign data as malicious. Therefore, higher values of TPR and lower values of FPR are preferable.

Table 5. TPR and FPR results.

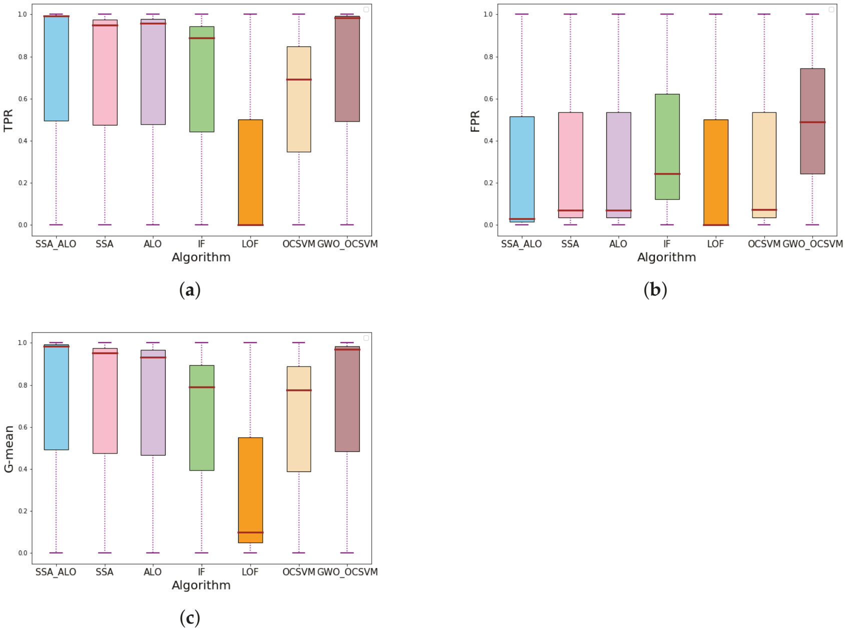
Device Type	SSA–ALO		SSA		ALO		IF [37]		LOF [37]		OCSVM [12]		GWO-OCSVM [12]	
	TPR	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**
Thermostat	**0.995**	0.002	0.993	0.003	0.991	0.002	0.992	0.101	0.001	**0.001**	0.498	0.139	0.960	0.009
Webcam	**0.999**	0.022	0.999	0.145	**0.999**	0.169	**0.999**	0.692	0.001	**0.001**	**0.999**	0.186	**0.999**	0.037
Baby monitor	0.989	0.002	0.948	0.049	0.977	0.061	0.618	0.003	0.004	0.004	0.234	**0.001**	**0.991**	0.016
Doorbell	**0.998**	0.068	0.972	0.120	0.982	0.097	0.826	0.010	0.001	**0.001**	0.923	0.003	0.995	0.083
Security camera	0.974	0.051	0.831	0.039	0.836	0.027	**0.999**	0.419	0.001	**0.001**	0.813	0.039	0.982	0.098
Average	**0.991**	0.029	0.949	0.071	0.957	0.071	0.887	0.245	0.002	0.002	0.693	0.074	0.985	0.489

In addition, and as seen in Table 6, SSA–ALO achieved the best G-mean values across all types of IoT devices. In this context, it is worth mentioning that the TPR and FPR could not clearly indicate the performance because the dataset was imbalanced. Therefore, we used the $G - mean$ measurement, which is commonly used to measure the performance of imbalanced datasets. The average G-mean value of the proposed SSA–ALO was 0.984, which is better than SSA, ALO, GWO-OCSVM, OCSVM, and IF. In addition, it is much higher than LOF. This means that SSA–ALO has a better ability to balance TPR and FPR.

Table 6. G-mean results.

Device Type	G-Mean						
	SSA–ALO	**SSA**	**ALO**	**IF** [37]	**LOF** [37]	**OCSVM** [12]	**GWO-OCSVM** [12]
Thermostat	**0.996**	0.995	0.994	0.944	0.032	0.655	0.975
Webcam	**0.988**	0.924	0.911	0.555	0.032	0.902	0.981
Baby monitor	**0.993**	0.949	0.958	0.785	0.063	0.479	0.987
Doorbell	**0.964**	0.925	0.942	0.904	0.032	0.959	0.956
Security camera	**0.961**	0.894	0.901	0.762	0.032	0.884	0.941
Average	**0.984**	0.952	0.931	0.789	0.099	0.776	0.968

To visually describe the comparisons between these algorithms, Figure 4 shows boxplots of the information in Tables 5 and 6. As shown in Figure 4, LOF has deficient performance since the TPR and FPR values are the same. This is close to the random classifier, where the TPR and FPR are equal. On the other hand, the proposed SSA–ALO shows different behavior, with a significant difference between the TPR and FPR values. This can be explained by the fact that the hybridization between SSA and ALO can optimize the FS problem better than other algorithms. Moreover, the feature subset found by the SSA–ALO is the best, so it results in the best classification results.



(**a**)

(**b**)

(**c**)

Figure 4. Boxplots to represent clearly the results in Tables 5 and 6. (**a**) TPR, (**b**) FPR, (**c**) G-mean.

The algorithms were also compared in terms of the running time of the algorithm (detection time) as seen in Table 7. It is clearly shown that the detection time of the SSA–ALO is the lowest among all the algorithms. On average, the SSA–ALO algorithm needs 5 s to detect an attack. This implies that if the IoT devices are attacked, then their functionality will be restored within 5 s. Thus, the time duration of the attack is reduced.

Table 7. Average time for detecting attack on NN-BaIoT dataset.

Device Type	**Average Detection Time (s)**						
	SSA–ALO	**SSA**	**ALO**	**IF** [37]	**LOF** [37]	**OCSVM** [12]	**GWO-OCSVM** [12]
Thermostat	**0.035**	0.390	0.420	3.325	1.107	0.087	0.047
Webcam	**0.091**	0.821	0.199	11.589	2.875	0.406	0.156
Baby monitor	**0.433**	0.749	0.552	49.698	33.317	1.889	0.622
Doorbell	**1.447**	1.963	2.331	18.208	7.753	1.570	1.099
Security camera	**22.659**	29.856	31.588	42.398	29.579	36.549	27.837

On the other hand, Figure 5 shows that the SSA–ALO algorithm obtains the best convergence behavior. Figures 6–9 present the radar charts, which demonstrate the overall performance results in terms of evaluation measures across the various IoT devices. It can be seen that the best feature subset that is found by the SSA–ALO algorithm enhances the results in terms of all the used evaluation measures. In summary, the proposed SSA–ALO demonstrated superior behavior for IoT attack detection in comparison with the other six anomaly detection algorithms.

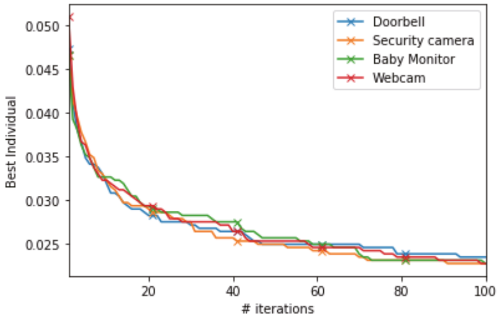


Figure 5. The convergence curves for SSA–ALO algorithm.

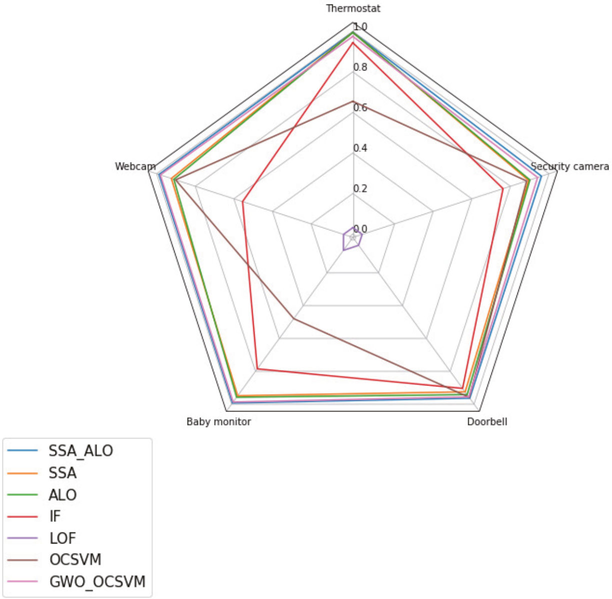


Figure 6. G-mean results for the SSA–ALO and other algorithms.

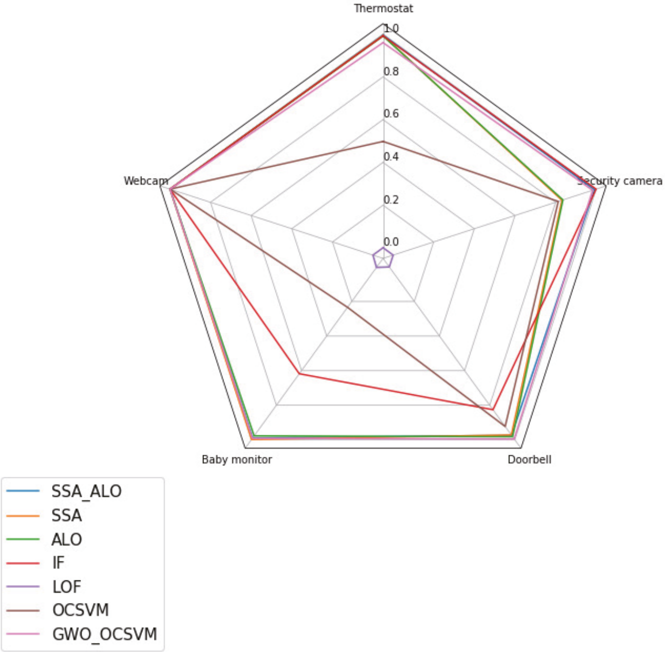


Figure 7. TPR results for the SSA–ALO and other algorithms.

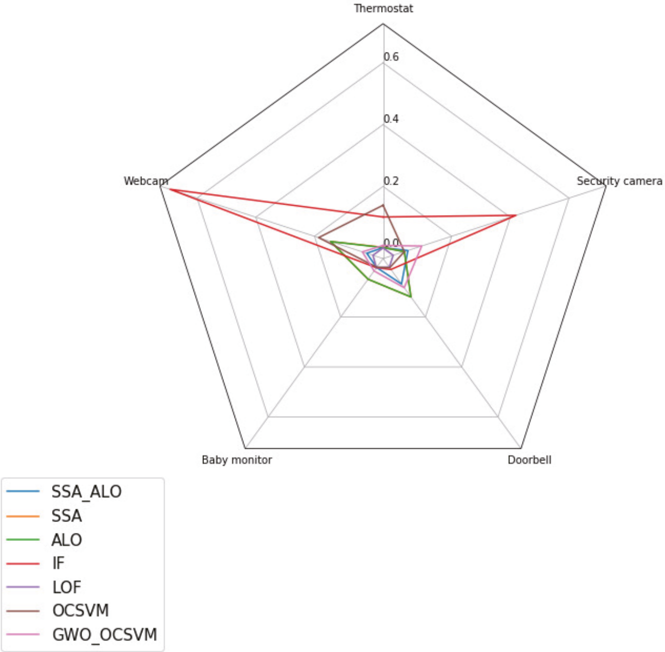


Figure 8. FPR results for the SSA–ALO and other algorithms.

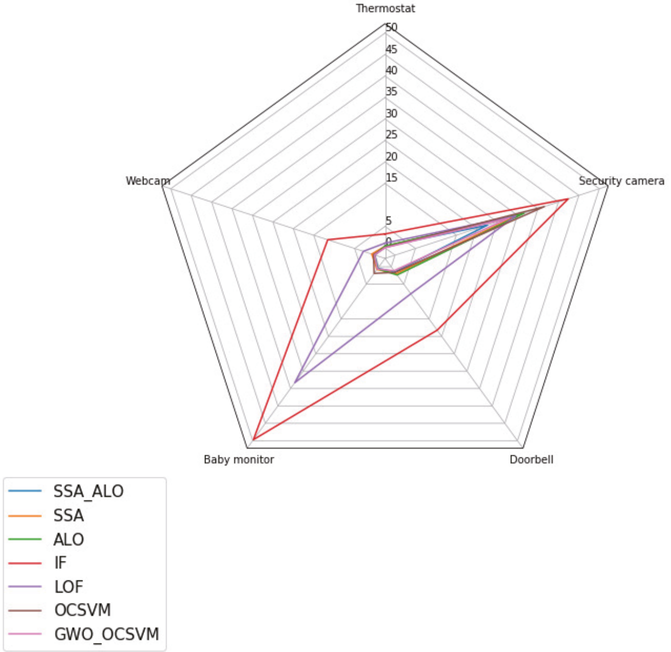


Figure 9. Time results for detecting attacks for the SSA–ALO and other algorithms.

6. Analysis of the Most Relevant Features

This section presents the most relevant features selected by the proposed SSA–ALO algorithm over the NN-BaIoT dataset during the optimization process. The results show that SSA–ALO obtained the best results in terms of the studied metrics using these relevant subsets of features. As mentioned before, the NN-BaIoT dataset has 115 features. The number of selected features by SSA–ALO to detect a threat to a security camera, baby monitor, doorbell, webcam, and thermostat is eleven, seven, twelve, eleven, and five, respectively. This implies that there is a dimensionality reduction of (90–96%). The following list shows the most relevant features necessary to detect an attack on each type of device:

- Security camera:
  1. H_L1_mean
  2. HH_L5_radius
  3. HH_L3_weight
  4. HH_L3_magnitude
  5. HH_L3_radius
  6. HH_L1_mean
  7. HH_L0.1_mean
  8. HH_L0.1_magnitude
  9. HH_L0.1_radius
  10. HH_iit_L0.1_variance
  11. HH_iit_L0.1_mean
- Baby Monitor:
  1. HH_L3_pcc
  2. HH_L0.1_magnitude
  3. HH_L0.1_radius
  4. HH_L0.1_pcc
  5. HH_iit_L1_weight

   6. HpHp_L3_magnitude
   7. HpHp_L3_radius
- Doorbell
   1. HH_L3_magnitude
   2. HH_L1_std
   3. HH_L1_radius
   4. HH_L0.1_weight
   5. HH_L0.1_mean
   6. HH_L0.1_pcc
   7. HH_L0.01_std
   8. HH_iit_L3_mean
   9. HH_iit_L1_weight
   10. HH_iit_L0.1_variance
   11. HpHp_L0.1_radius
   12. HpHp_L0.01_magnitude
- Webcam
   1. MI_dir_L0.1_mean
   2. MI_dir_L0.01_mean
   3. H_L5_variance
   4. H_L0.1_mean
   5. HH_L3_mean
   6. HH_L3_pcc
   7. HH_L0.1_pcc
   8. HH_L0.01_radius
   9. HpHp_L1_magnitude
   10. HpHp_L0.1_radius
   11. HpHp_L0.01_std
- Thermostat
   1. HH_L3_weight
   2. HH_L0.1_weight
   3. HH_L0.1_radius
   4. HH_iit_L1_weight
   5. HpHp_L3_mean

7. Conclusions and Future Work

IoT botnet attacks are brutal due to several reasons, such as the rapid increase in the number of connected IoT devices, the vulnerability of these objects to security breaches, and the fact that the attacked devices may not show any symptoms of threat. This study aims to detect IoT botnet breaches by utilizing ant lion optimization, salp swarm optimization, and the proposed hybrid SSA–ALO algorithm. The results on the NN-BaIoT dataset demonstrated the efficiency of the SSA–ALO. The new hybrid algorithm proved its efficiency compared to six other algorithms in terms of the studied evaluation metrics for all IoT device types under study. In addition, it needed the least time to detect an attack. Meanwhile, it reduced the number of selected features. The hybrid SSA–ALO algorithm has an excellent trade-off between the global search and the local search. The algorithm can increase the solutions' diversity and avoid premature convergence. For future work, we have a few ideas that can be investigated:

- Developing a parallel version of the hybrid SSA–ALO to work on a distributed framework.
- Using the proposed method on other applications such as those from genetic and microarray.

Author Contributions: These authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alieyan, K.; Almomani, A.; Abdullah, R.; Almutairi, B.; Alauthman, M. Botnet and Internet of Things (IoTs): A Definition, Taxonomy, Challenges, and Future Directions. In *Research Anthology on Combating Denial-of-Service Attacks*; IGI Global: Hershey, PA, USA, 2021; pp. 138–150.
2. Hamid, H.; Noor, R.M.; Omar, S.N.; Ahmedy, I.; Anjum, S.S.; Shah, S.A.A.; Kaur, S.; Othman, F.; Tamil, E.M. IoT-based botnet attacks systematic mapping study of literature. *Scientometrics* **2021**, *126*, 2759–2800. [CrossRef]
3. Dange, S.; Chatterjee, M. IoT Botnet: The Largest Threat to the IoT Network. In *Data Communication and Networks*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 137–157.
4. Viriyasitavat, W.; Da Xu, L.; Bi, Z.; Hoonsopon, D. Blockchain technology for applications in internet of things—Mapping from system design perspective. *IEEE Internet Things J.* **2019**, *6*, 8155–8168. [CrossRef]
5. Nguyen, H.T.; Ngo, Q.D.; Le, V.H. A novel graph-based approach for IoT botnet detection. *Int. J. Inf. Secur.* **2020**, *19*, 567–577. [CrossRef]
6. Qaddoura, R.; Al-Zoubi, A.; Almomani, I.; Faris, H. A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering with Oversampling. *Appl. Sci.* **2021**, *11*, 3022. [CrossRef]
7. Idrissi, I.; Boukabous, M.; Azizi, M.; Moussaoui, O.; El Fadili, H. Toward a deep learning-based intrusion detection system for IoT against botnet attacks. *IAES Int. J. Artif. Intell.* **2021**, *10*, 110.
8. Qaddoura, R.; M. Al-Zoubi, A.; Faris, H.; Almomani, I. A Multi-Layer Classification Approach for Intrusion Detection in IoT Networks Based on Deep Learning. *Sensors* **2021**, *21*, 2987. [CrossRef] [PubMed]
9. Aljarah, I.; Mafarja, M.; Heidari, A.A.; Faris, H.; Mirjalili, S. *Multi-Verse Optimizer: Theory, Literature Review, and Application in Data Clustering*; Nature-Inspired Optimizers: Berlin/Heidelberg, Germany, 2020; pp. 123–141.
10. Faris, H.; Aljarah, I.; Al-Madi, N.; Mirjalili, S. Optimizing the learning process of feedforward neural networks using lightning search algorithm. *Int. J. Artif. Intell. Tools* **2016**, *25*, 1650033. [CrossRef]
11. Mafarja, M.; Heidari, A.A.; Faris, H.; Mirjalili, S.; Aljarah, I. *Dragonfly Algorithm: Theory, Literature Review, and Application in Feature Selection*; Nature-Inspired Optimizers: Berlin/Heidelberg, Germany, 2020; pp. 47–67.
12. Al Shorman, A.; Faris, H.; Aljarah, I. Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 2809–2825. [CrossRef]
13. Khurma, R.A.; Aljarah, I.; Sharieh, A.; Mirjalili, S. Evolopy-fs: An open-source nature-inspired optimization framework in python for feature selection. In *Evolutionary Machine Learning Techniques*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 131–173.
14. Pamukov, M.E.; Poulkov, V.K.; Shterev, V.A. Negative selection and neural network based algorithm for intrusion detection in iot. In Proceedings of the 2018 41st International Conference on Telecommunications and Signal Processing (TSP), Athens, Greece, 4–6 July 2018; pp. 1–5.
15. Mehmood, A.; Mukherjee, M.; Ahmed, S.H.; Song, H.; Malik, K.M. NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks. *J. Supercomput.* **2018**, *74*, 5156–5170. [CrossRef]
16. Xiao, L.; Wan, X.; Lu, X.; Zhang, Y.; Wu, D. IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? *IEEE Signal Process. Mag.* **2018**, *35*, 41–49. [CrossRef]
17. Shaikh, F.; Bou-Harb, E.; Crichigno, J.; Ghani, N. A machine learning model for classifying unsolicited iot devices by observing network telescopes. In Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 25–29 June 2018; pp. 938–943.
18. Liu, L.; Xu, B.; Zhang, X.; Wu, X. An intrusion detection method for internet of things based on suppressed fuzzy clustering. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 1–7. [CrossRef]
19. Rathore, S.; Park, J.H. Semi-supervised learning based distributed attack detection framework for IoT. *Appl. Soft Comput.* **2018**, *72*, 79–89. [CrossRef]
20. Moustafa, N.; Turnbull, B.; Choo, K.K.R. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet Things J.* **2018**, *6*, 4815–4830. [CrossRef]
21. Hamamoto, A.H.; Carvalho, L.F.; Sampaio, L.D.H.; Abrão, T.; Proença, M.L., Jr. Network anomaly detection system using genetic algorithm and fuzzy logic. *Expert Syst. Appl.* **2018**, *92*, 390–402. [CrossRef]

22. Bin Ahmad, M.; Akram, A.; Asif, M.; Ur-Rehman, S. Using genetic algorithm to minimize false alarms in insider threats detection of information misuse in windows environment. *Math. Probl. Eng.* **2014**, *2014*, 179109. [CrossRef]
23. Hajimirzaei, B.; Navimipour, N.J. Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. *ICT Express* **2019**, *5*, 56–59. [CrossRef]
24. Ali, M.H.; Al Mohammed, B.A.D.; Ismail, A.; Zolkipli, M.F. A new intrusion detection system based on fast learning network and particle swarm optimization. *IEEE Access* **2018**, *6*, 20255–20261. [CrossRef]
25. Selvakumar, B.; Muneeswaran, K. Firefly algorithm based feature selection for network intrusion detection. *Comput. Secur.* **2019**, *81*, 148–155.
26. Panigrahi, A.; Patra, M.R. A Layered Approach to Network Intrusion Detection Using Rule Learning Classifiers with Nature-Inspired Feature Selection. In *Progress in Computing, Analytics and Networking*; Springer: Berlin/Heidelberg, Germany 2018; pp. 215–223.
27. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Netw.* **2013**, *11*, 2661–2674. [CrossRef]
28. Li, J.; Zhao, Z.; Li, R.; Zhang, H. Ai-based two-stage intrusion detection for software defined iot networks. *IEEE Internet Things J.* **2018**, *6*, 2093–2102. [CrossRef]
29. Xue, Y.; Jia, W.; Zhao, X.; Pang, W. An evolutionary computation based feature selection method for intrusion detection. *Secur. Commun. Netw.* **2018**, *2018*, 2492956. [CrossRef]
30. Popoola, E.; Adewumi, A.O. Efficient Feature Selection Technique for Network Intrusion Detection System Using Discrete Differential Evolution and Decision. *IJ Netw. Secur.* **2017**, *19*, 660–669.
31. Guendouzi, W.; Boukra, A. GAB-BBO: Adaptive Biogeography Based Feature Selection Approach for Intrusion Detection. *Int. J. Comput. Intell. Syst.* **2017**, *10*, 914–935. [CrossRef]
32. Gharaee, H.; Hosseinvand, H. A new feature selection IDS based on genetic algorithm and SVM. In Proceedings of the 2016 8th International Symposium on Telecommunications (IST), Tehran, Iran, 27–28 September 2016; pp. 139–144.
33. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
34. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]
35. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-baiot—Network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Comput.* **2018**, *17*, 12–22. [CrossRef]
36. Asuncion, A.; Newman, D. *UCI Machine Learning Repository*; University of California: Irvine, CA, USA, 2007.
37. Domingues, R.; Filippone, M.; Michiardi, P.; Zouaoui, J. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognit.* **2018**, *74*, 406–421. [CrossRef]

applied sciences

MDPI

Article

Research on Alarm Reduction of Intrusion Detection System Based on Clustering and Whale Optimization Algorithm

Leiting Wang, Lize Gu * and Yifan Tang

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; wlt562502@bupt.edu.cn (L.W.); tyfcs@bupt.edu.cn (Y.T.)
* Correspondence: glzisc@bupt.edu.cn

Abstract: With the frequent occurrence of network security events, the intrusion detection system will generate alarm and log records when monitoring the network environment in which a large number of log and alarm records are redundant, which brings great burden to the server storage and security personnel. How to reduce the redundant alarm records in network intrusion detection has always been the focus of researchers. In this paper, we propose a method using the whale optimization algorithm to deal with massive redundant alarms. Based on the alarm hierarchical clustering, we integrate the whale optimization algorithm into the process of generating alarm hierarchical clustering and optimizing the cluster center and put forward two versions of local hierarchical clustering and global hierarchical clustering, respectively. To verify the feasibility of the algorithm, we conducted experiments on the UNSW-NB15 data set; compared with the previous alarm clustering algorithms, the alarm clustering algorithm based on the whale optimization algorithm can generate higher quality clustering in a shorter time. The results show that the proposed algorithm can effectively reduce redundant alarms and reduce the load of IDS and staff.

Keywords: intrusion detection system; whale optimization algorithm; alarm reduction; hierarchical clustering

Citation: Wang, L.; Gu, L.; Tang, Y. Research on Alarm Reduction of Intrusion Detection System Based on Clustering and Whale Optimization Algorithm. *Appl. Sci.* **2021**, *12*, 11200. https://doi.org/10.3390/app112311200

Academic Editor: Ming-Chin Chuang

Received: 31 October 2021
Accepted: 19 November 2021
Published: 25 November 2021

1. Introduction

With the continuous development of computer network technology, people are more and more dependent on the convenience brought by the internet, but at the same time, the characteristics of the network, such as openness and complexity, also lead to the complexity and diversity of network security threats. In order to avoid the damage caused by network threats, many network security technologies are widely used, such as firewall, intrusion detection system (IDS), vulnerability scanning program and so on [1]. This study mainly focuses on the IDS, especially on how to improve the efficiency and performance of the IDS when dealing with network security events.

IDSs can be divided into two categories: the signature-based IDS and anomaly-based IDS [2]. The signature-based IDS determines whether network traffic shows malicious or normal behavior by maintaining a knowledge base [3]. The anomaly-based IDS detects whether the network traffic deviates from the normal rule state to determine malicious traffic [4]. Whether the signature-based IDS or the anomaly-based IDS can identify different types of network attacks is an important factor to judge its effectiveness. Therefore, the establishment of an intrusion detection system needs a network data set as the support. In past studies, many open network data sets were used by scholars as benchmark data sets, such as KDDCup99 [5] and NSL-KDD [6], which were widely used in various studies in the field of network security. However, with the rapid development of network technology and the emergence of new cyber security threats, these data sets have become outdated. In recent years, many new network data sets have been published on the internet, such as DDoS 2016 [7], UNSW-NB15 [8] and CICIDS 2017 [9]. Scholars are gradually using these relatively new data sets in their studies. Moreover, network data sets are still attracting the attention of scholars, such as LITNET-2020 [10], a new data set proposed by Damasevicius et al. based on the real network environment in 2020. These data sets usually have a fairly high-dimensional number of features, and different features may have different types, such as numerical type and categorical type. Due to the size of the data set, it is inevitable that there will be missing values in the data set. In the past, researchers proposed a series of methods, such as clustering, to deal with this problem [11–13].

According to literature statistics [14,15], the IDS will generate a large number of alarms in a very short period of time, 85% of which are irrelevant alarms or false alarms. In the past studies, many scholars have used different technologies to deal with the problem of redundant alarms generated by the IDS [16]. These methods can be generally divided into clustering-based methods [17–19], attribute-similarity-based methods [1,20], expert-system-based methods [21,22], genetic-algorithm-based methods [23,24], data-mining-based methods [25,26], etc.

Swarm intelligence optimization algorithms in recent years, as a kind of heuristic algorithm, are receiving more and more attention from researchers [27]. This kind of optimization algorithm is a good way to deal with the NP problem. The whale optimization algorithm (WOA), as an emerging swarm intelligence optimization algorithm, was proposed by Mirjalili and Lewis in 2016 [28]. Mirjalili and Lewis took inspiration from the behavior of humpback whales as they hunted their prey and modeled the process in the abstract into concrete mathematical equations. WOA is applied in many academic fields and achieves good results [29]. The specific application and theoretical background of WOA are described in detail in Sections 2 and 3.

The main contributions and findings of this paper are as follows:

- To deal with the alarm reduction problem, we propose a coding and decoding scheme that applies WOA to hierarchical clustering and propose a new fitness function. We apply crossover and mutation operators to WOA to enhance the search capability of the algorithm.
- To solve the problems of premature convergence of clustering and the tendency of clustering algorithm to fall into the local optimum, we propose a local version of WOA applied to hierarchical clustering, namely WOAHC-L. On the basis of WOAHC-L, we further propose a global version of WOAHC to resolve the problem of the high overlap degree of the cluster center, namely WOAHC-G.
- We conducted experiments on UNSW-NB15 data set to explore the performance of WOAHC in the search of cluster centers, time consuming, clustering results, accuracy and other indicators. Compared with the alarm hierarchical clustering algorithm in the past, the proposed framework can obtain higher quality alarm clustering within the allowed time range and solves the problem of alarm redundancy well.

The structure of this paper is as follows: The second part introduces the related work. The third part provides the theoretical background and introduces the framework of hierarchical clustering and the method of alarm distance calculation. In the fourth part, we propose our new methods for alarm hierarchy clustering, named WOAHC-L and WOAHC-G. The fifth part carries on the experiment and provides the experiment result and our discussion. The sixth part is the conclusion of this paper.

2. Literature Review

In the previous section, we gave an overview of several categories of methods for dealing with redundant alarms. In this section, we mainly discuss hot approaches for dealing with alarm problems in recent years and explore the application of heuristic algorithms (such as WOA) in dealing with alarm problems in the intrusion detection domain. Firstly, we introduce the research results of scholars on alarm problems of the past few years. Wang et al. [30] proposed a framework to improve the intelligent false alarm reduction for DIDS based on edge computing devices. They built a false alarm filter by using machine learning classifiers, which can select an appropriate algorithm to maintain the filtration accuracy. Toldinas et al. [31] proposed a new image recognition method using multi-level deep learning to solve the problem of intrusion detection system identification of network attacks. They converted network features into four-channel images that were used to train and test the pre-trained deep learning model ResNet50. Kinghorst et al. [32] introduced a pre-processing step in the process of alarm flood analysis to enhance the robustness of the alarm system in dealing with the random alarm or interference alarm mode through probability calculation of alarm correlation. Fahimipirehgalin et al. [33] proposed a data-driven method, using alarm log files to detect the causal sequence of alarms. In this method, an efficient alarm clustering method based on the time distance between alarms is proposed, which is helpful to preserve adjacent alarms in a cluster. To solve the problem of a large number of redundant alarms generated by IDS, Sun and Chen [1] proposed an alarm aggregation scheme based on the combination of conditional rough entropy and knowledge granularity. Based on this scheme, the weights of different attributes in the alarms were obtained, and the similarity values of the alarms were calculated within the sliding time window to aggregate the similar alarms to reduce redundant alarms.

In recent years, the development of swarm intelligence optimization algorithms has attracted the attention of researchers. Swarm intelligence (SI) optimization algorithms can be divided into two main categories: one is the particle swarm optimization algorithm (PSO), and the other is the ant colony optimization algorithm (ACO). The emergence of SI was first used to solve optimization problems and was subsequently applied by scholars in the field of network attack detection. Alharbi et al. [34] proposed a method combining the bat algorithm and neural network to detect botnet attacks. The bat algorithm is used to select feature subsets and adjust hyperparameters in a network attack, and is used to adjust the hyperparameters and weight optimization of a neural network. In article [35], Khurma et al. combined the salp swarm algorithm and ant lion optimization algorithm to propose a wrapper feature selection model to solve the problem of high dimension of features in IDS. Zhang et al. [36] proposed an improved particle swarm optimization algorithm to solve the problems of repeated alarms and high false positive rate in IDS. In the process of reconstructing the attack path between DDoS attack victims and attackers based on an internet protocol backtracking scheme, Lin et al. [37] proposed a multi-mode optimization scheme that applied the improved locust swarm optimization algorithm to the reconstructed attack path in order to solve the problem that the traditional route search algorithm was prone to fall into local optimum. This method shows the excellent search performance of the SI algorithm. In addition, there is also a lot of research of SI in the feature selection stage of the IDS and attack target detection [38–40].

PSO and ACO algorithms have achieved good results in many fields. On this basis, scholars have proposed more excellent swarm intelligence optimization algorithms inspired by nature, such as the WOA [28], bat algorithm [41], wolf optimization algorithm [42], pathfinder algorithm [43], etc. Mirjalili and Lewis studied the behavior of humpback whales in preying on prey, analyzed and modeled the behavior patterns of the bubble net attack and spiral approach, and put forward the WOA. It is proved that the WOA has strong competitiveness, compared with the existing meta-heuristic algorithms and traditional algorithms. After WOA was proposed, due to its excellent problem optimization ability, it was quickly applied in various fields of research. In a review article on the application of WOA [29], the author listed the research progress of WOA, including hybridization, improvement and variation, as well as application scenarios such as engineering problems, clustering problems, classification problems, image processing, network and task scheduling and other problems. It can be seen that WOA, as a new meta-heuristic swarm intelligent optimization algorithm, has proved its reliability and good performance in handling optimization problems. However, in the field of alarm clustering, previous scholars did not carry out further research on it. Based on the proven global and local search capabilities of WOA, this paper studies the application of WOA in alarm clustering, focusing on the optimization of alarm hierarchical clustering based on WOA.

3. Theoretical Background

In this section, we introduce the relevant theoretical background of the study. Firstly, we introduce the alarm reduction algorithm based on hierarchical clustering, including the concept of generalization level, the calculation method of distance and the basic process of the algorithm. Second, we introduce the main ideas and basic process of WOA. Table 1 shows the list of notations used in this paper.

Table 1. Table of notations used in this paper.

Symbol	Description
N_i	ith node of a hierarchical tree
A_i	ith alarm in an alarm set
D_{Ai-Aj}	The distance between ith alarm and jth alarm in an alarm set
C_i	ith alarm cluster center
$\vec{X}(t)$	In WOA, the current solution at iteration t
$\vec{X}^*(t)$	In WOA, the best solution at iteration t
$\vec{X}_{rand}$	In WOA, a random solution in the current solution space
$\vec{D}$	In WOA, the distance between ith $\vec{X}$ and jth$\vec{X}$
$SearchAgents$	In WOA, the number of search agents to search solution simultaneously
$MaxIter$	In WOA, a predetermined maximum number of iterations
C, r, a, l	Random numbers used in WOA to control logical judgment
$E_Tk(t, x)$	Fitness value of alarm number for cluster center k
$E_s(k)$	Fitness value of alarm distance for cluster center k
$O\left(C_i, C_j\right)$	The degree of overlap between ith cluster center C_i and jth cluster center C_j
ES_O	Fitness value of the coincidence degree of all cluster centers in the cluster
TP	Number of normal network traffic clustering to normal cluster
TN	Number of network attack clustering to attack cluster
FP	Number of network attack alarms incorrectly clustering to normal cluster
FN	Number of normal network traffic incorrectly clustering to attack cluster

3.1. Alarm Reduction Algorithm Based on Hierarchical Clustering

3.1.1. Generalization Hierarchies

We first introduce the concept of generalization hierarchy. As mentioned earlier, if newly generated alarms are arranged in a meaningful cluster according to predefined rules, operators can easily understand what is happening in the network. According to this idea, we define the concept of cluster, and classify the alarms into the cluster they belong to according to the rules. We use the basic idea of hierarchical clustering proposed by Julisch [44,45]. As shown in Figure 1 below, for all the attributes in the alarm, we can use the method of hierarchical division to layer the attributes. Figure 1a shows the attribute hierarchical tree composed of IP attributes, and each leaf node of the tree represents a unique specific IP address. We can generalize it once to obtain the specific protocol using this IP, such as firewall and WWW/FTP in Figure 1a. If we continue to generalize it, we can obtain more advanced generalizations, such as DMZ and EXTERN. When we find that the generalization has reached the highest level and can no longer be generalized, we define the root of the hierarchy tree. For example, the root of the hierarchy tree to which the IP attribute belongs is ANY IP. The generalized structure of other attributes is similar, as shown in Figure 1b–d. In the past, scholars have proposed many methods for the construction of a hierarchical tree, with which we can construct a hierarchical tree for various attributes of the alarm data.

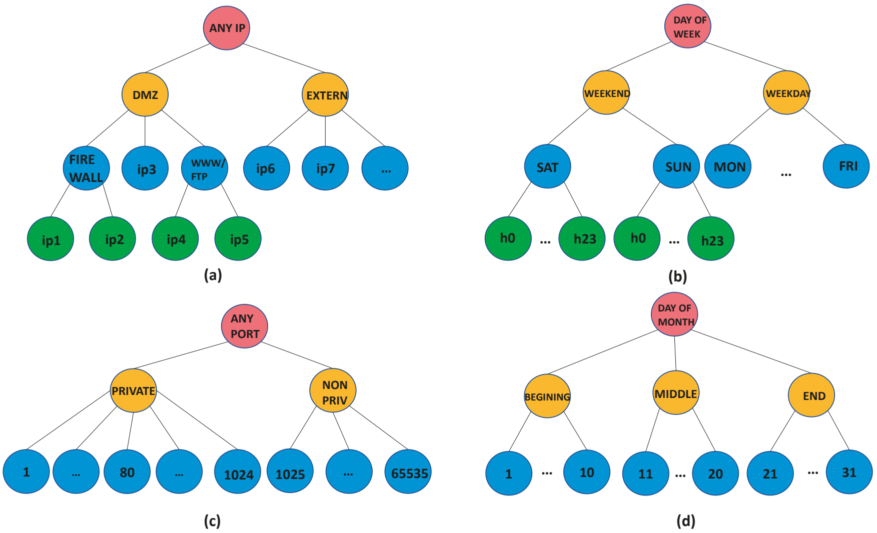


Figure 1. Hierarchical tree structure of four attributes: (**a**) IP address attribute; (**b**) time attribute measured in weeks; (**c**) port number attribute; (**d**) time attribute measured in months.

After providing the construction process of a hierarchical tree, we provide the following definitions of nodes in the hierarchical tree.

Definition 1. *A basic alarm is the alarm triggered by IDS that correspond to leaf nodes in the hierarchy tree. An abstract alarm is derived from the basic alarm by generalization and corresponds to intermediate or root nodes in the hierarchy number. Naturally, a basic alarm is also a special abstract alarm.*

Definition 2. *In a hierarchical tree, if there is a path from N_1 node to N_2 node, then N_1 is a generalization of N_2, and N_2 is a specification of N_1.*

Definition 3. *For both abstract alarms A_1 and A_2, A_1 is a generalization of A_2 if each attribute in A_1 is a generalization of the corresponding attribute in A_2, and at the same time, A_2 is a specification of A_1.*

Definition 4. *For an alarm set, the minimum cover refers to the common generalization of all alarms in the set, and the generalization is a minimum specification.*

Based on the four definitions above, considering the four hierarchical trees shown in Figure 1, there is an alarm set that contains three alarms: A_1 (ip1,80,h1,11), A_2 (DMZ,80,h0, MIDDLE), and A_3 (DMZ, PRIVATE, WEEKEND, MIDDLE). A_1 is a basic alarm because all the attributes of the alarm are at leaf nodes in the hierarchical tree. A_2 and A_3 are abstract alarms because there is at least one attribute in the alarm that is the middle nodes in the hierarchy tree. A_3 is a generalization of A_1 and A_2 because every attribute of A_3 is a generalization of A_1 and A_2, and obviously A_3 is a common generalization of A_1, A_2, and A_3.

3.1.2. Distance Definition

After obtaining the generalized alarm set, in order to cluster the alarms in the original alarm set, we need to define the distance calculation rule in the clustering problem, that is, defining the distance between two alarms to judge whether they belong to the same cluster. In fact, it is easy to calculate the distance between attributes of a numeric type, but there is a problem if the alarm property is a category, time, or string property using the same distance calculation method. We give the following definition to calculate the distance between two alarms in a hierarchical tree.

Definition 5. *The distance between any two nodes in the same hierarchical tree depends on the number of edges between them. If two nodes have directly linked edges, the distance between them is 1.*

Definition 6. *If there is a generalization–specification relationship between two alarms, the distance between the two alarms is defined as the average distance between their attributes.*

Definition 7. *The distance of an alarm set is defined as the average distance between the minimum coverage in the set and each alarm.*

Consider the alarm sets A_1 (ip1,80,h1,11), A_2 (DMZ,80,h0,MIDDLE), and A_3 (DMZ, PRIVATE, WEEKEND, MIDDLE) mentioned above, where the minimum coverage in the alarm set is A_3. The distance between A_1 and A_2 is (2 + 0 + 2 + 1)/4 = 1.25. The distance between the minimum coverage and alarm sets is (1.5 + 0.75 + 0)/3 = 0.75.

3.1.3. Definition of the Clustering Problem

The clustering method is now described as the following: among all triggered alarms, a group of generalized alarms is found; the number of alarms within each generalized alarm exceeds or is equal to a given threshold; and the distance between the alarms is as small as possible. This method is proved to be an NP complete problem, that is, the exact solution cannot be obtained in feasible time. Julisch presented an approximate algorithm [44] as shown in Algorithm 1.

Algorithm 1 Julisch's alarm hierarchical clustering algorithm

```
Input: a set of events; a threshold T; a set of trees of all the attributes considerer;
Output: an alarm/ /an abstract event
1: select an arbitrary alarm A, each member of which is a leaf in a tree
2: while the number of events A covers is less than T do
3:         select an arbitrary member of A, and replace the member with its direct parent
4: end while
```

3.2. Whale Optimization Algorithm

Mirjalili and Lewis proposed the whale optimization algorithm based on abstract modeling of the hunting strategies of humpback whales; it mimics the bubble-net feeding in the foraging behavior of humpback whales [28]. Humpback whales hunt close to the surface while trapping the prey in a net of bubbles. They create this net when swimming on a '6′-shaped path. The algorithm mimics two phases: the first phase (exploitation phase) is to encircle the prey and attack with spiral bubble nets, and the second phase (exploration phase) is searching randomly for prey. Figure 2 shows a series of behaviors of humpback whales as they hunt prey. Figure 2a shows the movement of the whale toward the prey, during which the whale can choose to move toward the lead whale or in a random direction. Figure 2b illustrates the shrinking encircling mechanism used by whales to capture prey. Besides the shrinking encircling mechanism, the whale also moves further toward the prey in a spiral shape, during which the whale emits a bubble attack to surround the prey, as shown in Figure 2c. The details of each phase are presented in the following subsections.

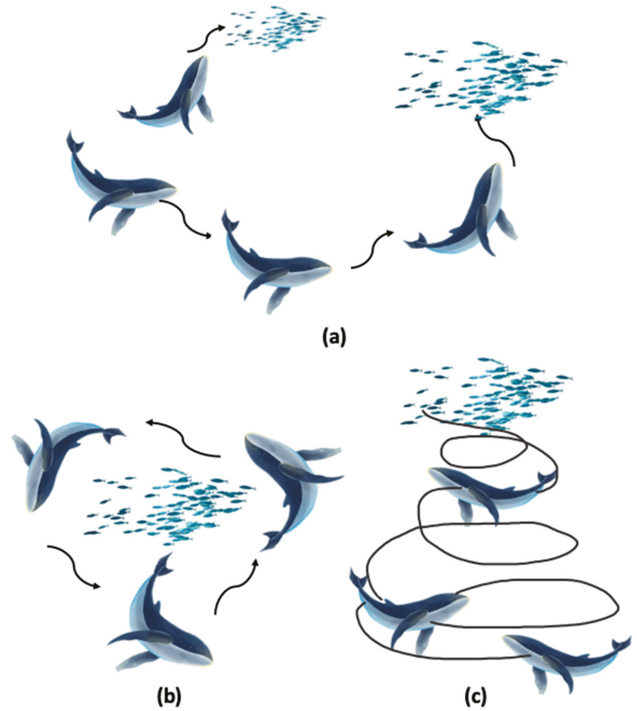


Figure 2. The process of humpback whales hunting prey: (**a**) the whales move toward the lead whale or in random directions; (**b**) whales reach their prey by the shrinking encircling mechanism; (**c**) whales reach their prey in a spiral shape.

3.2.1. Exploitation Phase (Encircling Prey/Bubble-Net Attacking Method)

Mirjalili et al. designed two methods to mathematically model the bubble-net behavior of humpback whale, one of which is the shrinking encircling mechanism and the other is the spiral updating position. We then analyze the concrete implementation of these two processes from a mathematical point of view.

In the shrinking encircling mechanism, WOA applies the following two formulas to update the problem solution to model the movement of a whale toward a prey.

$$\vec{D} = \left|\vec{C}.\vec{X}^*(t) - \vec{X}(t)\right| \tag{1}$$

$$\vec{X}(t+1) = \left|\vec{X}^*(t) - \vec{A}.\vec{D}\right| \tag{2}$$

where t represents the number of current iterations, $\vec{X}^*$ represents the optimal solution obtained so far, $\vec{X}$ is the current solution scheme, $||$ is the absolute value, and . is the dot product operation between the elements. $\vec{A}$ and $\vec{C}$ are the coefficient vectors, which can be obtained from Equations (3)–(5):

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{3}$$

$$\vec{C} = 2.\vec{r} \tag{4}$$

$$a = t\frac{2}{MaxIter} \tag{5}$$

where the value of A decreases linearly from 2 to 0 and its value is in the interval $[-a, a]$. r is a random vector between [0, 1]. a increases linearly from 0 to 2 depending on the number of iterations. t is the number of the current iteration, and $Maxiter$ is the maximum number of pre-set iterations.

According to Equation (2), the current solution updates the position of the current solution according to the optimal solution obtained so far. Through the two vectors A and C, the search range of the current solution can be controlled to be fixed within the neighborhood range of the optimal solution. In order to imitate the behavior of whales hunting prey in Figure 2b, we use the mathematical model shown in Figure 3a for modeling and analysis. It is assumed that (X^*, Y^*) is the current global optimal solution, and the solid dots in the figure, such as (X, Y), are the current solution. Figure 3a shows the possible positions from (X, Y) toward (X^*, Y^*) that can be achieved by $0 \leq A \leq 1$ in a 2D space.

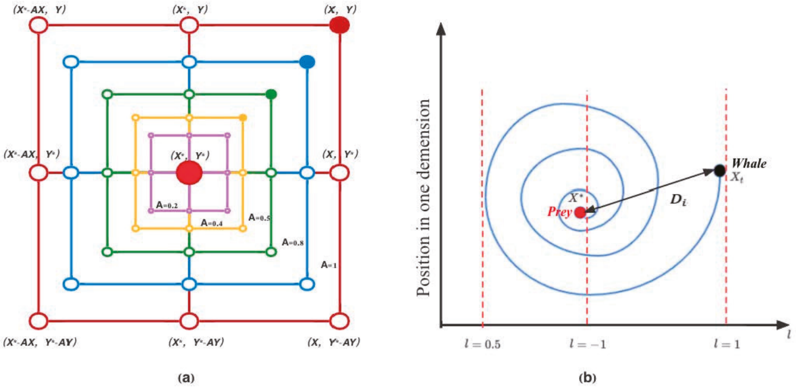


Figure 3. Bubble-net search mechanism implemented in WOA. (X^* represents the best solution obtained so far): (**a**) shrinking encircling mechanism; (**b**) spiral updating position.

As mentioned above, whales also use a spiral motion to move toward prey as shown in Figure 2c. WOA uses the following formula to model this behavior.

$$\vec{X}(t+1) = D'.e^{bl}.cos(2\pi l) + \vec{X}^*(t) \tag{6}$$

where $\vec{X}^*$ represents the optimal solution obtained so far, $\vec{X}$ is the current ith solution, D' = $\mid \vec{X}^*(t) - \vec{X}(t) \vee$ and indicates the distance of the ith whale to the prey (best solution obtained so far), b is a constant for defining the shape of the spiral, and l is a random variable between $[-1, 1]$.

The approximate figure of Equation (6) and Figure 2c is shown in Figure 3b. In this 1D space, X_t represents the current ith solution (i.e., the whale), X^* represents the current optimal solution (i.e., the prey), and the distance between X_t and X^* is D_i. The x-coordinate of the coordinate axis represents a random number l, which is used to control the movement direction of the whale, and the y-coordinate represents the next position $X_{(t+1)}$ of the current solution X_t. In order to simulate the behavior of humpback whales swimming around prey while following a spiral-shaped track in a shrinking circle, the authors consider the contraction and spiral rise processes to occur equally with probability, and the mechanism is defined in Equation (7).

$$\vec{X}(t+1) = \begin{cases} \left| \vec{X}^*(t) - \vec{A}.\vec{D} \right| if(p < 0.5) \\ D'.e^{bl}.cos(2\pi l) + \vec{X}^*(t) if(p \geq 0.5) \end{cases} \tag{7}$$

where p is a random variable between [0, 1].

3.2.2. Exploration Phase (Search for Prey)

As mentioned above, besides moving toward the lead whale, the whale can also move in a random direction, as shown in Figure 2a. This is called the exploration phase in WOA. In this phase, we no longer require a random search of the solution based on the position of the optimal solution found so far, but instead update the position with randomly selected solutions. Thus, a vector with a random value greater than 1 or less than −1 is used to force a solution away from the optimal search agent. This mechanism can be expressed in mathematical models as Equations (8) and (9).

$$\overrightarrow{D} = \left|\overrightarrow{C}.\overrightarrow{X}_{rand} - \overrightarrow{X}\right| \tag{8}$$

$$\overrightarrow{X}(t+1) = \left|\overrightarrow{X}_{rand} - \overrightarrow{A}.\overrightarrow{D}\right| \tag{9}$$

where $\overrightarrow{X}_{rand}$ is a random solution of the current solution vector set. The meanings of the other notations are mentioned above.

In WOA, the author uses A to control whether the algorithm specifically executes the exploitation phase or exploration phase. When the absolute value of A is greater than 1, WOA chooses to execute the exploration phase; when the absolute value of A is less than 1, WOA chooses to execute the exploitation phase. As mentioned in the previous paper, the value range of A is $[-a, a]$, and the value of A decreases linearly with the increase in the number of iterations. Therefore, in the general trend, WOA has more chances to jump out of the current optimal solution and choose the random solution at the early stage of implementation. With the increase in the number of iterations, the range of A will gradually shrink, and the WOA will gradually converge to the optimal solution.

4. Proposed Method

In this section, we introduce our proposed algorithm in detail. First, in Section 4.1, we introduce two different coding schemes corresponding to the local and global versions of the WOA applied to hierarchical clustering. In Section 4.2, we describe the fitness function that generates alarm clustering using the WOA. In Section 4.3, we combine the WOA with the crossover and variation factors of the genetic algorithm and propose the pseudo-codes of the local and global versions of the WOA alarm hierarchical clustering algorithm.

4.1. Encoding and Decoding

We first introduced the encoding and decoding scheme of the local version of the WOA applied to hierarchical clustering. In this scheme, a search agent in the WOA corresponds to a cluster center of hierarchical clustering. As mentioned above, a cluster center is composed of a basic alarm or an abstract alarm, so we can obtain the data structure encoded by the search agent. Each attribute in the alarm corresponds to a binary string in the encoded data structure, represented by 0 or 1, as shown in Figure 4.

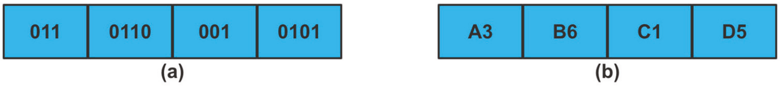


Figure 4. A cluster center (A3,B6,C1,D5) and its corresponding coding scheme: (**a**) the coding scheme of the cluster center; (**b**) the cluster center (A3,B6,C1,D5).

Figure 5a shows the coding scheme of the cluster center in binary form, and Figure 5b shows the attribute values of the cluster center corresponding to this coding. Figure 4 shows an alarm with four attributes (A, B, C, and D) with decimal values of 3, 6, 1, and 5.

The four attribute fields of the alarm are located in their respective hierarchical trees, as shown in Figure 5.

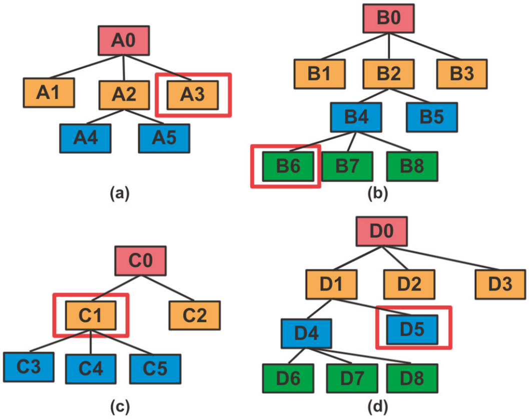


Figure 5. Hierarchical tree structure for attributes A, B, C and D. (**a**) three-layer tree structure of attribute A; (**b**) four-layer tree structure of attribute B; (**c**) three-layer tree structure of attribute C; (**d**) four-layer tree structure of attribute D.

We can find the location of the node corresponding to the binary-encoded attribute in the hierarchical tree. At the same time, we can easily obtain the binary fragment corresponding to the alarm through the hierarchical tree. This coding scheme indicates that a search agent corresponds to a cluster center. The goal of the WOA is to find the best search agent for the fitness function over multiple iterations, output its corresponding cluster center, and categorize the alarms that belong to that cluster.

In this encoding and decoding scheme, a WOA search agent corresponds to a cluster center, assuming that the cluster center is composed of N attributes and the binary length of each attribute is K, then the encoding length of a search agent is $N * K$, corresponding to N hierarchical trees.

After giving the encoding and decoding scheme of the local version WOA–hierarchical tree, we introduce the encoding and decoding schemes of the global version WOA–hierarchical tree. In the coding scheme of the global version, a WOA search agent is composed of a group of cluster centers. Assuming that WOA eventually obtains C cluster centers, each of which is composed of N attributes with length K, the coding length of the global version WOA–hierarchical tree is $C * N * K$. Let us take the example shown in Figure 6 for illustration.

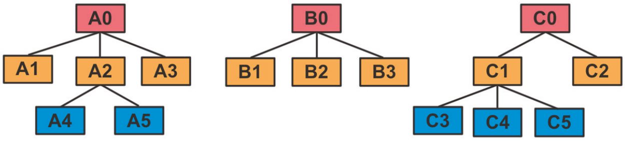


Figure 6. Hierarchical tree structure for attributes A, B and C.

In Figure 6, there are three hierarchical trees corresponding to the three attributes of an alarm, respectively. If we need to finally obtain three cluster centers for this alarm set, which are $(A4, B2, C3)$, $(A1, B1, C2)$ and $(A3, B2, C4)$, then one of our search agents can be encoded as shown in Figure 7a in the second coding method. Figure 7b shows the three cluster centers corresponding to this coding scheme.

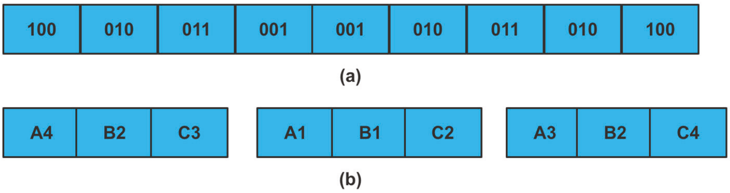


Figure 7. Three cluster centers and its corresponding coding scheme: (**a**) the coding scheme of the three cluster centers; (**b**) three cluster centers (A4,B2,C3), (A1,B1,C2) and (A3,B2,C4).

4.2. Fitness Function

The core of WOA is to find the best solution set in a finite solution set space through a finite number of iterations. The fitness function is the standard to evaluate whether a solution set is excellent. Therefore, how to set up an appropriate fitness function is the key to solve the problem of hierarchical clustering using WOA. The selection of the fitness function in this paper mainly considers the following three factors: the number of alarms contained in the cluster center, the distance between alarms belonging to the same cluster, and the coincidence degree between clusters. We believe that, given a fixed threshold of alarm distance, the more alarms that a cluster contains, the greater the fitness value will be. In addition, when the similarity of alarms belonging to the same cluster is higher (the distance is smaller), the fitness value is higher. If the coincidence degree of cluster center is higher, we believe that the meanings of the two clusters are closer, and the overall fitness value will be smaller.

For a given alarm cluster center, $S = (N_1, N_2, N_{3,}N_4, \ldots N_m)$, where the value of m corresponds to the number of hierarchical trees used in the alarm cluster. If the alarm distance meets the number of alarms within the given threshold, we believe that the fitness of the alarm cluster center is higher. In Refs. [45–47], the setting of the fitness function is to determine whether the number of alarms belonging to a certain alarm cluster center exceeds the given threshold. If the number exceeds, the fitness is set to 1, and if not, the fitness is set to 0. This processing method has a simple idea and can well distinguish the alarms that do not meet the clustering requirements from those that meet the clustering requirements. However, the problem is that the method cannot reflect the quality of the cluster centers which exceed the threshold value. For example, if the threshold value is set to 500, the existing two clusters C_1 and C_2 contain alarm numbers of 2000 and 5000, respectively. We intuitively feel that C_2 is better than C_1 but their fitness values are set to the same value, which does not achieve a good distinction. In this paper, a new calculation method of alarm number fitness is adopted, as shown in Equation (10).

$$E_T k(t,x) = \begin{cases} 0, if(x < t) \\ ln(\frac{x}{t}), if(x \geq t) \end{cases} \tag{10}$$

where t represents the threshold of the number of alarms that the cluster should contain, and x represents the number of alarms belong to the cluster center.

When $x < t$, we think that the cluster contains too few alarms, and the cluster center should not be selected; when $x > t$, we think that the fitness value of the cluster center increases with the increase in x, and taking $t = 500$ as an example, the image of the fitness function is shown in Figure 8.

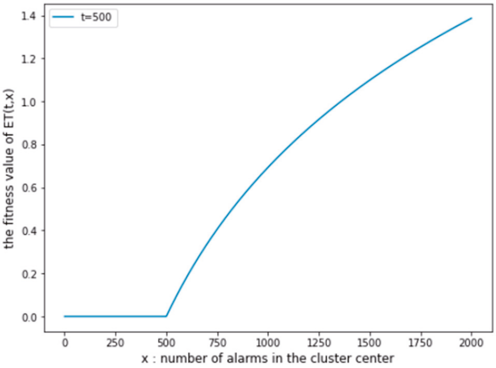


Figure 8. Figure of fitness function $E_Tk(t,x)$ for t = 500.

For a cluster center, only considering the number of alarms contained in the cluster center as an evaluation index cannot indicate the quality of the cluster center. Only when the number of alarms contained in the cluster center is large enough and the difference between alarms is small enough do we believe that the selection of the cluster center is reasonable. Therefore, we define a fitness function for the internal differences in the alarm cluster center, as shown in Equation (11). The average depth of the four hierarchical trees as shown in Figure 1 is (3 + 2 + 3 + 2)/4 = 2.5.

$$E_S(k) = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{D(C_i)}{M_d}\right) \tag{11}$$

where $i = (1, 2, 3, \ldots, n)$ represents n alarms belonging to a cluster center k, $D(C_i)$ represents the sum of the distances between the alarm and each attribute of the cluster center in its attribute tree, and M_d represents the average depth of all attribute hierarchy trees.

4.3. Crossover and Mutation Operator

One of the difficulties of the WOA in solving hierarchical clustering problems is how to apply Equations (2), (6) and (9) to transform search agent positions for different types of attributes. If an attribute is a continuous variable, the use of the above formula is not affected, but if an attribute is a discrete variable then using the formula is difficult. Because we use the attributes of the hierarchical tree structure and type of binary coding structure, we can easily transform the attributes into a hierarchical tree. Here, we use crossover and mutation operators of the genetic algorithm to solve this problem. Another advantage of using these two operators is that the WOA is combined with crossover and genetic operators to further improve the algorithm's ability to search for local and global optimal solutions. This conclusion is mentioned in Ref. [48].

Now, we present the application of crossover operator based on the WOA coding scheme. Taking an alarm with four attributes as an example, the binary identity of the attribute field with two alarms is shown in Figure 9.

As can be seen from Figure 10, the attributes of the two coded alarms are $Alarm1$ $(A_3B_6C_1D_5)$ and $Alarm2$ $(A_3B_{10}C_2D_3)$. Starting with 6 bit, cross transposition of the attributes of the two alarms can be carried out so that two new alarms can be obtained after operation, as shown in Figure 10.

From Figure 10, we can see that the two new alarms, $Alarm1'$ $(A_3B_6C_2D_3)$ and $Alarm2'$ $(A_3B_{10}C_1D_5)$, are generated after crossing. Looking at the changes in the attribute fields of the two alarms, we find that, except for the change in the attribute of the exchange location, the other attributes only changed the location.

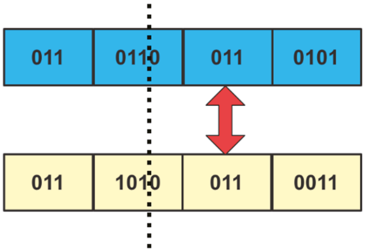


Figure 9. Property fields for the two alarm records waiting for a crossover operation.

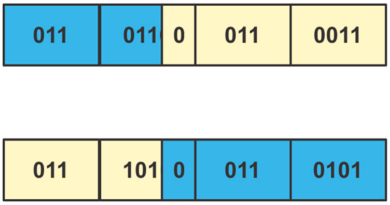


Figure 10. Two new alarm records after a crossover operation.

After introducing the application of crossover operator in alarm clustering, we introduce the use of the mutation operator in alarm clustering. Taking the alarm shown in the left figure of Figure 9 as an example, the change of the alarm after a mutation operation is performed on a bit of the alarm attribute is shown in the right figure of Figure 11. By changing the value of the sixth bit in the binary from 1 to 0, the alarm changes from $Alarm\ (A_1, B_5, C_6)$ to $Alarm(A_1, B_4, C_6)$. Observing the change in the alarm, we can find that the mutation operation only makes a certain attribute of the alarm field change, while the other attributes remain unchanged.

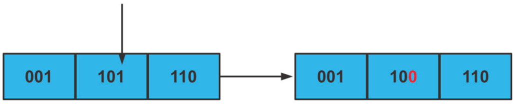


Figure 11. An alarm record before/after mutation operation.

4.4. WOA-Based Alarm Hierarchical Clustering Process

After introducing the coding and decoding scheme, fitness function and crossover and mutation operator of the WOA applied to hierarchical clustering, in this section, we present the processing of the local and global versions of WOA applied to the alarm hierarchical clustering process. In order to express clearly, the WOA hierarchical clustering of the local version and global version are respectively called WOAHC-L and WOAHC-G. Compared with the traditional alarm hierarchical clustering algorithm, which can only generate one random generalization alarm at a time, WOA uses multiple search agents to search the solution set space of the generalization alarm simultaneously, which can improve efficiency and obtain more possibilities of solution sets. The random agent selection stage of WOA provides a higher possibility to jump out of the local optimum to find a better solution set. The use of crossover operators and mutation operators can help WOA deal with various types of data and enhance the ability of local search and global search.

We first provide the algorithm flow of WOAHC-L. The process of WOAHC-L can be described as follows: the algorithm first initializes several search agents, each representing a cluster center of the alarm. Then, we calculate the fitness value of each search agent according to the fitness function. The cluster center represented by the search agent with the best fitness value is the optimal solution. After that, according to WOA's search mechanism, the remaining search agents explore the solution set space around the optimal solution through exploitation phase and exploration phase and update the value of the optimal solution whenever there is a better solution. After several iterations, the cluster center represented by the search agent with the optimal fitness value is output, and alarms belonging to that cluster are added to the cluster and removed from the original alarm set. Each time the algorithm is executed, a cluster center is output and the alarms belonging to the cluster are deleted from the original alarm set. When the remaining alarms no longer meet the clustering rules after several times of algorithm execution, the algorithm is finished. The alarm clustering process based on WOAHC-L is shown in Figure 12.

The algorithm pseudocode of WOAHC-L is shown in Algorithm 2.

Algorithm 2 WOA for alarm hierarchical clustering (local version)

```
Input: threshold, N, maxIteration, mutation rate, t = 0, k = 0
Output: alarm_center[], alarms after clustering
while the number of current alarms is bigger than threshold do
    Randomly generate initial population Xi (i = 1, 2, . . . , N)
    Calculate the fitness value of each solution
    Get the best Xi that has the largest fitness value, mark it as X*
    while $t < MaxIterationr$ do
        for population Xi(i = 1, 2, . . . , N) do
            Use Eqations (3)–(5) to update a,A,C
            Generate values for random numbers 1 and P
            if $p < 0.5$ then
                if $|A| < 1$ then
                    Use Equations (1) and (2) to update the position of Xi
                    Apply mutation operation on X* (best solution) given mutation rate (r) to get Xmut
                    Perform crossover operation between Xmut and Xi
                    Set the new position of Xi to the output of crossover operation
                else if $|A| \geq 1$ then
                    Select a random search agent (Xrand)
                    Use Equations (8) and (9) to update the position of Xi
                    Apply mutation operation on Xrand given mutation rate(r) to get Xmut
                    Perform crossover operation between Xmut and Xrand
                    Set the new position of Xi to the output of crossover operation
                end if
            else if $p \geq 0.5$ then
                Use Equation (6) to update the position of the current solution Xi
                Apply mutation operation on X* (best solution) given mutation rate (r) to get Xmut
                Perform crossover operation between Xmut and Xi
                Set the new Position of Xi to the output of crossover operation
            end if
        end for
        Calculate the fitness value of each solution, check if any solution goes beyond the search space
        If there is a better solution Xi update Xi as X*
        t = t + 1
    end while
    Mark X* as cluster k
    Put alarms which belong to the cluster k to the alarm_center[k]
    Remove alarms from the original alarms set
    k = k + 1
end while
```

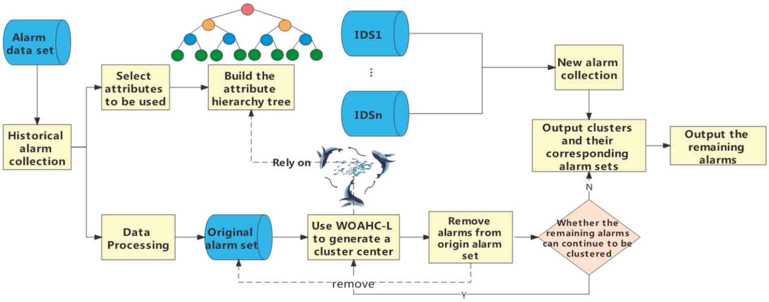

Alarm data set
Historical alarm collection
Select attributes to be used
Build the attribute hierarchy tree
Rely on
IDS1
⋮
IDSn
New alarm collection
Output clusters and their corresponding alarm sets
Output the remaining alarms
N
Data Processing
Original alarm set
Use WOAHC-L to generate a cluster center
Remove alarms from origin alarm set
Whether the remaining alarms can continue to be clustered
remove
Y

Figure 12. Alarm reduction framework flow chart based on WOAHC-L.

Based on the excellent local and global search capabilities and the group search mechanism of WOAHC-L algorithm, we can find excellent alarm cluster centers. However, the problem of WOAHC-L is that there may be a high degree of overlap between the cluster centers obtained by executing WOAHC-L several times, so that the alarm sets originally belonging to the same cluster may be divided into multiple clusters. In other words, WOAHC-L only focuses on the generation of single cluster centers and does not consider the overlap between the finally obtained cluster centers. In order to solve the above problem of the WOAHC-L algorithm, we propose a global version based on WOA hierarchical clustering, namely WOAHC-G, which uses the second encoding scheme of the search agent mentioned in Section 4.1 above.

The process of WOAHC-G can be described as follows: the algorithm initializes several search agents, each of which is composed of N cluster centers and represents the final clustering result. According to the fitness function, each search agent calculates a fitness value. The search agent with the best fitness value is the optimal search agent, and the cluster center represented by the agent is the final cluster center set. Based on the WOA exploitation phase and exploration phase, the cluster center set represented by the search agent with the optimal fitness value is finally obtained after several iterations. WOAHC-G differs from WOAHC-L in that WOAHC-G only needs to execute once to obtain all cluster centers, while WOAHC-L needs to execute several times until the number of remaining alarms is insufficient for the next algorithm execution. In addition, WOAHC-G considers the problem of coincidence degree between different cluster centers and takes the coincidence degree as an important indicator of fitness value. Therefore, the fitness function of the algorithm needs to be changed to add the coincidence degree of cluster centers, as shown in Equation (12).

$$\boldsymbol{O}(\boldsymbol{C_i}, \boldsymbol{C_j}) = \begin{cases} \boldsymbol{1}, \boldsymbol{C_i} \cap \ \boldsymbol{C_j} = \varnothing \\ \boldsymbol{0}, \boldsymbol{C_i} \cap \ \boldsymbol{C_j} \neq \varnothing \end{cases} \tag{12}$$

where $\boldsymbol{C_i}, \boldsymbol{C_j}$ represent two cluster centers. If each attribute of the two cluster centers has no intersection, the coincidence degree is considered to be 0; otherwise, it is 1.

After the calculation function of clustering coincidence degree is given, the evaluation equation of clustering coincidence degree is given as Equation (13).

$$ES_o = \frac{2}{k(k-1)} \sum_{0 \le i < j \le k} O(C_i, C_j) \tag{13}$$

where k represents the number of cluster centers. When there are k cluster centers, $\frac{k(k-1)}{2}$ times are needed to calculate the coincidence degree between them. Therefore, the coefficient in the calculation formula of ES_o is set as $\frac{2}{k(k-1)}$ to ensure that the value of ES_o is within the interval [0, 1].

The alarm clustering process based on WOAHC-G is shown in Figure 13.

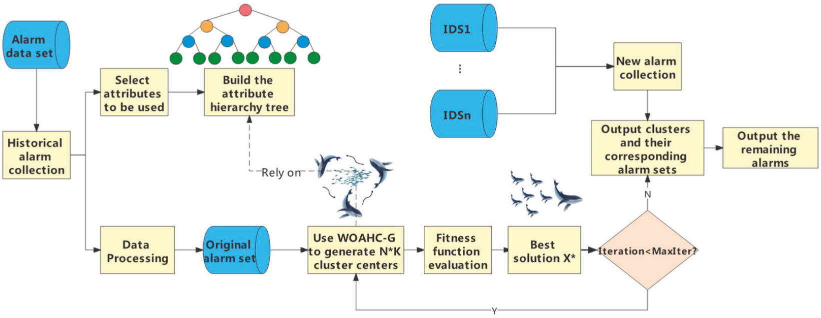

Alarm data set
Historical alarm collection
Select attributes to be used
Build the attribute hierarchy tree
Rely on
Data Processing
Original alarm set
Use WOAHC-G to generate N*K cluster centers
Fitness function evaluation
Best solution X*
Iteration<MaxIter?
N
Y
IDS1
⋮
IDSn
New alarm collection
Output clusters and their corresponding alarm sets
Output the remaining alarms

Figure 13. Alarm reduction framework flow chart based on WOAHC-G.

The algorithm pseudocode of WOAHC-G is shown in Algorithm 3

Algorithm 3 WOA for alarm hierarchical clustering (global version)

```
Input: N, maxIteration, mutation rate, t = 0, k = 0
Output: alarm_center[], alarms after clustering
Randomly generate initial population Xi (i = 1, 2, . . . , N)
Calculate the fitness value of each solution
Get the best Xi that has the largest fitness value, mark it as X*
while $t < MaxIteration$ do
      for population Xi (i = 1, 2, . . . , N) do
         Use Equations (3)–(5) to update a,A,C
         Generate values for random numbers 1 and p
         if $p < 0.5$ then
            if $|A| < 1$ then
                 Use Equations (1) and (2) to update the position of Xi
                 Apply mutation operation on X* (best solution) given mutation rate (r) to get Xmut
                 Perform crossover operation between Xmut and Xi
                 Set the new position of Xi to the output of crossover operation
             else if $|A| \geq 1$ then
                 Select a random search agent (Xrand)
                 Use Equations (8) and (9) to update the position of Xi
                 Apply mutation operation on Xrand given mutation rate (r) to get Xmut
                 Perform crossover operation between Xmut and Xrand
                 Set the new position of Xi to the output of crossover operation
             end if
           else if $p \geq 0.5$ then
              Use Equation (6) to update the position of the current solution Xi
              Apply mutation operation on X* (best solution) given mutation rate (r) to get Xmut
              Perform crossover operation between Xmut and Xi
              Set the new position of Xi to the output of crossover operation
         end if
         end for
         Calculate the fitness of each solution, check if any solution goes beyond the search space
         If there is a better solution Xi, update Xi as X*
         t = t + 1
end while
return X*
```

5. Experiments and Results

5.1. Experiment Data Set

In this section we describe the data sets used in the experiment. We use UNSW-NB15 as the experimental data set [8]. The UNSW-NB15 data set was developed by Ixia Perfectstorm. It is used to simulate and generate real and contemporary attack models. This is a tool called Tcpdump, which contains up to 100 GB of PCAP files and is used to simulate nine different types of attacks. These include DOS, Shellcode, worms, Fuzzers, backdoors, exploits, analytics, generality, and scouts. In addition, the data set consists of 12 algorithms for generating 49 features belonging to class tags. The following Table 2 shows a set of features in UNSW-NB15, along with the corresponding groups and data types.

Table 2. UNSW-NB15 Features with their data type and category.

Category	No	Name	Data Type	Category	No	Name	Data Type
Flow	1	srcip	Nominal	Content	25	trans_depth	Integer
	2	sport	Integer		26	res_bdy_len	Integer
	3	dstip	Nominal	Time	27	Sjit	Float
	4	dsport	Integer		28	Djit	Float
	5	proto	Nominal		29	Stime	Timestamp
Basic	6	state	Nominal		30	Ltime	Timestamp
	7	dur	Float		31	Sintpkt	Float
	8	sbytes	Integer		32	Dintpkt	Float
	9	dbytes	Integer		33	Tcprtt	Float
	10	sttl	Integer		34	Synacj	Float
	11	dttl	Integer		35	Ackdat	Float
	12	sloss	Integer	General	36	Is_sm_ips_ports	Binary
	13	dloss	Integer	Purpose	37	Ct_state_ttl	Integer
	14	service	Nominal		38	Ct_flw_http_mthd	Integer
	15	Sload	Float		39	Is_ftp_login	Binary
	16	Dload	Float		40	Ct_ftp_cmd	Integer
	17	Spkts	Integer	Connection	41	Ct_srv_src	Integer
	18	Dpkts	Integer		42	Ct_srv_dst	Integer
Content	19	swin	Integer		43	Ct_dst_ltm	Integer
	20	dwin	Integer		44	Ct_src_ltm	Integer
	21	stcpb	Integer		45	Ct_src_dport_ltm	Integer
	22	dtcpb	Integer		46	Ct_dst_sport_ltm	Integer
	23	smeansz	Integer		47	Ct_dst_src_ltm	Integer
	24	dmeansz	Integer		48	Attack_cat	Nominal
					49	Class	Binary

The UNSW-NB15 data set has a total of 2,540,044 records, which are stored in four files respectively. In order to better conduct the experiment, the data set provides the training set and test set that have removed the missing values, with 175,341 records and 82,332 records respectively. As can be seen from the above table, the 49 fields in the data set include fields of different types, such as Flow, Basic, Content, Time, Content, etc., and each attribute belongs to either the discrete or continuous types.

5.2. Experimental Setup

The experimental operating environment used Intel Core i5-7500CPU 3.40 GHz and 8 GB memory. The configuration of the software environment is as follows: the operating system is Microsoft Windows 10, the experimental program is written in Python, and the development version is Python 3.7.3. We use PyCharm and Juptyer Notebook as the integrated development environment for Python. In addition, we use WEKA as an auxiliary tool for data processing and analysis. WEKA is an open-source machine learning and data mining software based on Java environment [49]. It is one of the most complete data mining tools today.

5.3. Experimental Results

As mentioned above, we use the UNSW-NB15 data set as our experimental data set. A total of 50,801 data were randomly selected from the training set and test set to carry out the clustering calculation of the alarm. The extracted alarm label distribution is shown in Table 3.

Table 3. Distribution of alarm labels in 50,801 pieces of data.

	Attack_Catgory	Number		Attack_Catgory	Number
1	Null	44,387	6	Reconnaissance	289
2	Generic	4257	7	Backdoor	53
3	Exploits	913	8	Analysis	52
4	Fizzers	463	9	Shellcode	29
5	DoS	355	10	Worms	3

Attack types marked 2–10 in the table represent specific types of attacks, while the type marked 1 is normal network traffic. It is not difficult to see that this data set contains a large number of normal network traffic, which is identified by IDS as a malicious alarm, resulting in a large number of false alarms. Our goal is to use clustering methods to correctly identify normal network traffic and eliminate it from alarms.

We analyze the source IP address and destination IP address in the experimental data set as follows. The data set contains two types of IP addresses: Class B and Class C (such as Class C IP addresses 149.171.126.43 and 149.171.126.50). We can extract the same fields and classify the IP addresses, and the uncertain part is represented by X. The statistics of each type of IP address and its number are shown in Table 4.

Table 4. Distribution of source and destination IP addresses.

Source_IP	Number	Destination_IP	Number
59.166.0.X	38,817	149.171.126	46,051
175.45.176.X	7234	175.45.176	4284
149.171.126.X	4338	10.40.X.X	298
10.40.X.X	412	224.0.0.X	108
		59.166.0.X	55
		192.168.241	5

After analyzing the experimental data, we first explore the performance of WOAHC-L in looking for a single alarm cluster center. The specific parameters of WOAHC-L are set as follows: $SearchAgents = 10$ and $MaxIter = 50$, respectively, which means that we use 10 search agents to search the solution space of the clustering at the same time, and the maximum number of iterations of the algorithm is 50. Other parameters in WOAHC-L, such as r, a, p, etc., are generated by random numbers or are related to $MaxIter$ so we do not have to set it up. The fitness function is set as the equation mentioned in Section 4.2.

According to the mechanism of WOAHC-L, the cluster center obtained by calling the algorithm for the first time has the best fitness value. In order to explore the performance of WOA in hierarchical clustering, we independently perform WOAHC-L four times and record the change in the optimal fitness value with the number of iterations during each algorithm execution. The performance of WOAHC-L is shown in Figure 14. The X axis is the number of iterations and the Y axis is the fitness value of the currently found optimal cluster center.

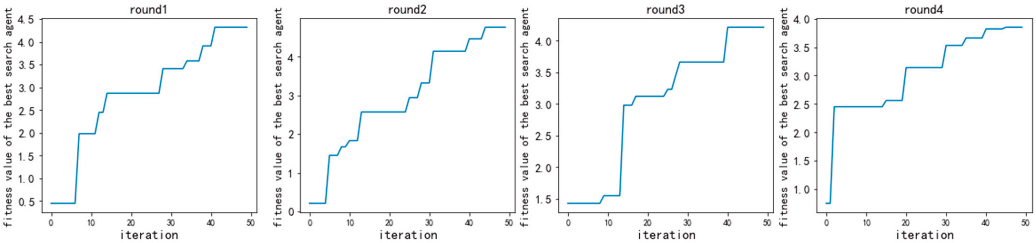


Figure 14. Comparison of fitness values of the best search agent obtained by four calls to WOAHC-L.

As can be seen from Figure 14, WOAHC-L constantly seeks the most excellent cluster center in the search space through its exploitation phase and exploration phase. It can be seen from the figure that WOAHC-L has an excellent ability to break through the local optimum, which can be clearly seen from the stage of 20–50 iteration times.

After verifying that WOAHC-L has excellent global search capability in alarm clustering, we continue to explore the use of WOAHC-L to solve the local clustering problem of alarm hierarchical clustering, that is, every time we call the WOAHC-L, an alarm cluster is obtained, which is obtained through the iterative search of WOAHC-L, according to the fitness function. Taking $MaxIter$ of 50 as the maximum number of iterations and the number of search agents of 10 as an example, we perform WOAHC-L four times and obtain four cluster centers successively from 50,801 alarms. The number of alarms contained in the four cluster centers varies with the number of iterations as shown in Figure 15.

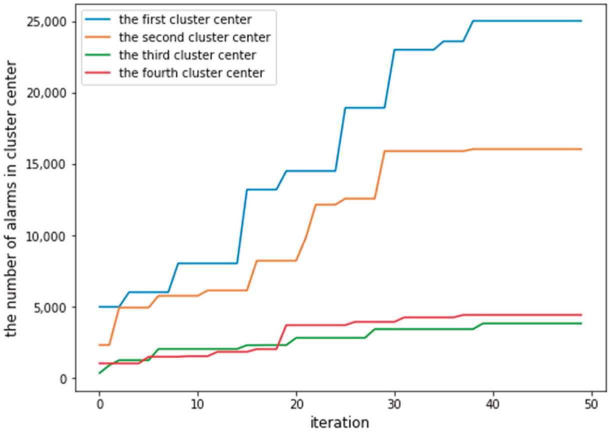


Figure 15. Alarm clustering result of 4 consecutive calls to WOAHC-L.

According to the experimental statistics, the number of alarms contained in each cluster obtained by calling WOAHC-L four times is as follows. By analyzing the obtained cluster center, we find that the tag field in the cluster center obtained by the first, second and fourth calls to WOAHC-L is $(null, 0)$, and the tag field in the cluster center obtained by the third calls to WOAHC-L is $(Generic, 1)$. In other words, these four calls to WOAHC-L distinguish the generic attacks from the normal traffic in the original data set by clustering. We add up the number of alarms generated by the first, second and fourth calls to WOA to 45,469, which is within the allowable range when compared with the number of normal network traffic 44,387 in the table. The third call to the WOAHC-L cluster is 3819, which is within an acceptable distance of the 4257 alarms for the generic attacks in the table. This experiment proves that hierarchical clustering using WOAHC-L can well distinguish the type of alarm.

After proving WOAHC-L's excellent local and global search ability and clustering ability, we next explore statistical analysis of WOAHC-L on time consumption and clustering results. Based on the algorithm flow of WOAHC-L in Figure 12, when the number of remaining alarms no longer meets the clustering threshold, the algorithm ends, and the clustering result is output. Therefore, we constantly call WOAHC-L to obtain new clustering until the condition of the algorithm ending is met. The experimental results are shown in Figure 16.

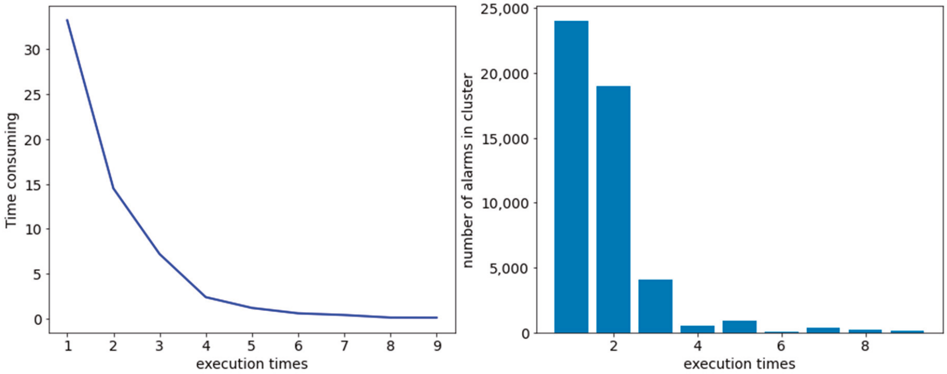


Figure 16. Time consuming and alarm clustering results versus the execution times of WOAHC-L.

As can be seen from Figure 16, as the number of algorithm executions increases, the time of each algorithm execution gradually decreases, and the number of alarms contained in the cluster obtained by each invocation of the algorithm also decreases. The reason is obviously that every time the algorithm is called to obtain the cluster center, the alarms belonging to the cluster in the alarm set are removed from the alarm set, so the next time the algorithm is called, the alarm items scanned by the algorithm are fewer and fewer, and the time and number of alarms are also reduced.

In order to verify the robustness of WOAHC-L, we conduct five experiments. In each experiment, WOAHC-L is called several times to obtain multiple cluster centers, until the amount of remaining data in the data set are less than the requirement of alarm clustering.

As can be seen from Table 5, except the first row, there are five rows in the table, representing five experiments. Except the first column, from a total of 12 columns, the top 11 columns represent each experiment that invokes the WOAHC-L to obtain cluster center containing the number of alarms. What we need to pay attention to is that in the table there are some gaps, such as the third line 10 columns; these blanks show that the experiments, after several WOAHC-L calls, have met the conditions of the end of the algorithm, such as the second experiment. We find that WOAHC-L is called eight times and then stops executing. The last column of the table represents the total time of the experiment. Compared with the five experiments, the average time and standard deviation of the total time spent in our calculation algorithm are 58.7 s and 5.3, which are within the allowable range of the experiment. In addition, we should also note that although the number of cluster centers generated by each experiment is different (the five experiments are 11, 8, 10, 10, and 9, respectively), as we mentioned before, cluster centers generated by this kind of local clustering method will cause the problem of a too high coincidence degree of cluster centers. However, we can study the experimental data in the first experiment of two previous cluster centers containing the sum of the number of alarms for 33,303. For the properties of alarm analysis, we find that they all belong to the normal network traffic, so we can combine the two-cluster center as a cluster center, representing the clustering of normal network traffic. Excluding the problem of clustering overlap, we have reason to believe that the robustness of the algorithm is relatively excellent.

Meanwhile, in order to compare with other algorithms, we repeat the alarm clustering algorithm mentioned in the references [45–47] and compare it with WOAHC-L. In order to more intuitively feel the results of clustering, we use the number of alarms contained in the cluster center as the Y axis to compare the four algorithms. Please note that it is not accurate to rely only on the number of clusters contained in the cluster center to evaluate the quality of the clustering results. The quality of the clustering results is also greatly related to the distance between the alerts contained in the cluster center (called difference in some works), that is, the fitness function mentioned in the formula. The experimental results are shown in Figure 17.

Table 5. The results of the number of alarms contained in the cluster and the number of algorithms calls in five experiments.

Time Rounds	1	2	3	4	5	6	7	8	9	10	11	Time Consuming
1	18,433	14,870	3219	6553	4170	745	421	921	31	29	51	66.3 s
2	38,923	4331	3490	993	671	210	78	32				52.0 s
3	36,544	4370	4921	2008	719	198	43	23	104	97		56.5 s
4	24,003	18,995	4109	529	899	32	390	241	45	122		55.3 s
5	15,023	30,901	2104	2233	920	320	429	290	102			63.4 s

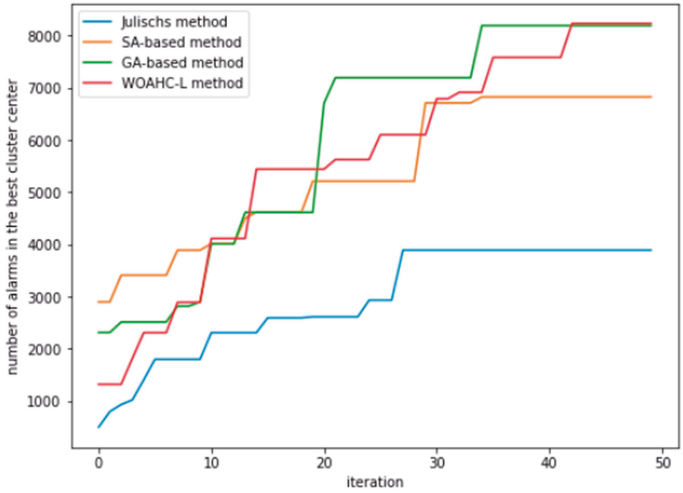


Figure 17. Comparison of experimental results between WOAHC-L and Julisch's method, SA and GA.

As can be seen from Figure 17, compared with Julisch's algorithm, SA and GA, WOAHC-L can well jump out of the local optimum in the process of clustering search, while other algorithms begin to converge at the beginning or middle of the iteration and no longer search for a better cluster center. We need to emphasize that the purpose of this experiment is to intuitively show that compared with the other three algorithms, WOAHC-L can better jump out of the local optimal and achieve better results when carrying out hierarchical clustering. A more detailed comparison of experimental results with other algorithms is presented in the following experiments. We use the following formula to calculate the values of these indicators.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{14}$$

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

where TP represents the number of normal network traffic clustering to normal cluster, TN represents the number of network attack clustering to attack cluster, FP represents the number of network attack alarms incorrectly clustering to normal cluster, FN represents the number of normal network traffic incorrectly clustering to the attack cluster. $Accuracy$

represents the accuracy of the clustering; *Precision* represents how much of the data clustered as normal traffic are really normal traffic; and *Recall* represents how much of the normal traffic is correctly clustered.

Table 6 shows the comparison of various indicators of different algorithms. These indicators have specific meanings, introduced in Table 1.

Table 6. A detailed comparison of experimental results between WOA and Julisch's method, GA and SA.

	Cluster Numbers	**Time Consuming**	**Remaining Alarms**	**Accuracy**	**Precision**	**Recall**
Julisch's method	16	43.2	831.4	91.8%	97.6%	92.9%
GA-based	10	31.5	432.4	93.5%	97.1%	95.4%
SA-based	13	25.5	913.0	91.7%	96.7%	93.7%
WOAHC-L	10	68.3	322.5	95.2%	98.4%	96.1%

As can be seen from Table 6, WOAHC-L is obviously superior to other algorithms in terms of the number of clustering, number of remaining alarms, accuracy, precision and recall. It should be noted that recall refers to the rate of aggregation of false alarms, or reduction in redundant alarms. However, WOAHC-L has obvious deficiencies in terms of time consumption, which is caused by the multi-search agent mechanism of the WOA and the calculation of the fitness value.

As the excellent performance of WOAHC-L in alarm reduction is proved by comparing with other algorithms, we explore the stability of WOAHC-L in clustering and the rule analysis of clustering results. In order to avoid the contingency of experimental results, we conduct eight experiments, where each experiment performs several WOAHC-L times until the number of remaining alarms no longer meets the clustering requirements. The results of the experiment are shown in Figure 18. The X axis represents the number of cluster centers, that is, the number of WOAHC-L calls, and the Y axis represents the number of remaining alarms in the original alarm set.

According to Figure 18, we find that in the eight experiments, WOAHC-L generates cluster centers with more alarms in the previous calls. According to the output of the code, we can find that these cluster centers are all clusters related to redundant alarms, which is consistent with our expected assumption: WOAHC-L first clusters redundant false alarms that account for a higher proportion in the alarm data set, and then clusters the remaining real alarms. We set the alarm number threshold of algorithm end condition as 500 and observe that the times of calling WOAHC-L are different in the eight groups of experiments (10, 6, 10, 7, 8, 6, 10, and 7, respectively). In Section 4.4, we explained the reason for this, that is, there may be an overlap between the cluster centers generated by WOAHC-L, and it is confirmed in our experiment.

Through a series of experiments above, we prove the superiority of WOAHC-L in the process of alarm reduction, but its disadvantages are also exposed: the algorithm is time consuming, the results obtained by multiple executions of the algorithm have certain differences (also known as idempotency), and the high degree of overlap between some cluster centers lead to an increase in algorithm calls. To solve this problem, we propose a global clustering version of WOAHC, namely WOAHC-G, and explore the performance of WOAHC-G in the following experiments.

As mentioned in Section 4.4, WOAHC-G only needs to be called once to obtain all the cluster centers, provided that the number of cluster centers need to be specified, which is a difficult problem. However, with the experimental results of WOAHC-L as a reference, we can choose an appropriate number of cluster centers as a parameter. Here, we set the number of cluster centers as eight, which is based on the above experimental results of the comprehensive consideration. Meanwhile, our search agent code is 72 dimensions (each cluster center contains nine attributes, a total of eight cluster centers). We still select 10 search agents to search for the cluster space. Other parameters of WOAHC-G are set as above, and the fitness function of the global version is selected, adding the influence of clustering coincidence degree. Under the same data set configuration, we conduct eight experiments and use a boxplot to represent the number of alarms in each cluster, as shown in Figure 19.

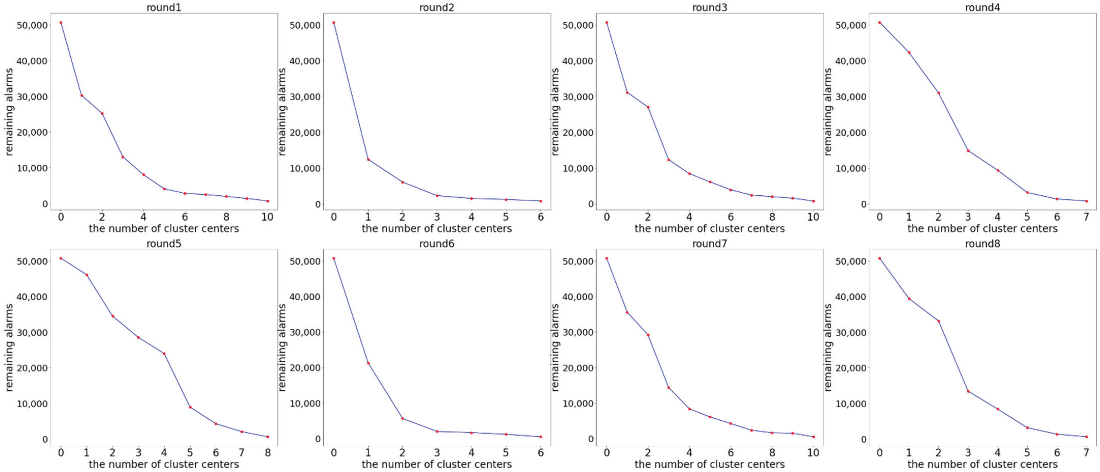


Figure 18. Comparison of the final results of eight experiments.

In each boxplot in Figure 19, the blue line at the top represents the maximum, the blue line at the bottom represents the minimum, the upper edge of the square represents the upper quartile, the lower edge represents the lower quartile, the red line represents the median, the green dot represents the average, and the white dots represent outliers. As can be seen from Figure 19, when we fix the number of clustering as 8, the experimental results generated by the eight experiments are basically not significantly different. As can be seen from the number of alarms in cluster 1 to cluster 3, the global clustering version using WOAHC-G can effectively eliminate the problem of excessive cluster repetition. However, the problem of the global clustering version is that it is difficult to determine an appropriate number of cluster centers. In this round of experiments, we set the number of cluster centers to 8 according to the experience of WOAHC-L to eliminate the impact of this problem. However, how to determine an appropriate number of cluster centers requires further research and analysis in the future.

Finally, we discuss the time complexity of the proposed algorithm framework. It is assumed that WOAHC eventually generates C clusters, with a total of M alarms to be clustered, the maximum iteration time of the algorithm is T, there are N search agents searching the solution set space at the same time, and each hierarchical tree has m nodes. For WOAHC-L, the time complexity of calculating the fitness value is $O(M + log_2 m)$, so the total time complexity of the algorithm is $O(C * T * N * (M + log_2 m))$. During our experiment, N threads are used to search N search agents at the same time, so the time complexity can be optimized to $O(C * T * (M + log_2 m))$. For WOAHC-G, the time complexity of calculating the fitness value is $O\left(log_2 m + M + C^2\right)$, so the overall time complexity is $O\left(T * N * \left(log_2 m + M + C^2\right)\right)$, and the time complexity after multithreading optimization is $O\left(T * \left(log_2 m + M + C^2\right)\right)$.

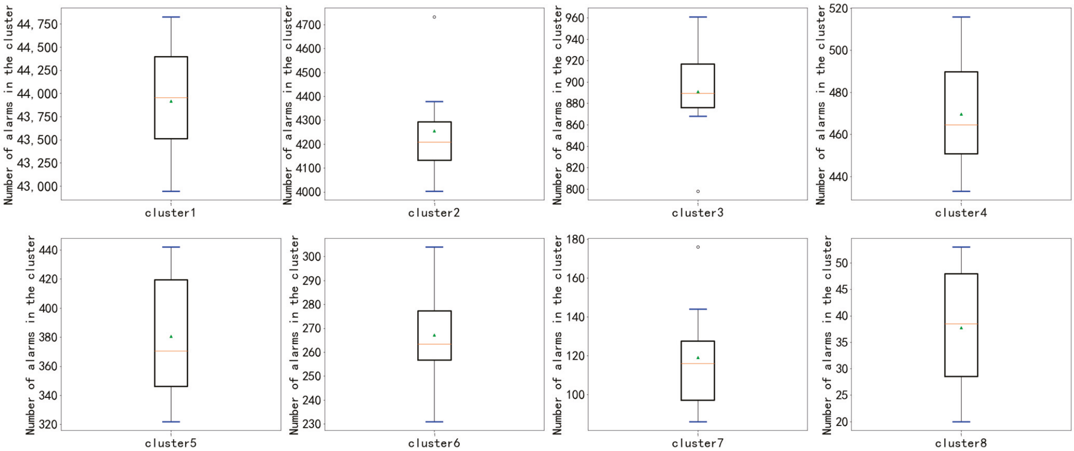


Figure 19. Boxplots of the number of alarms in 8 clusters.

6. Conclusions and Future Work

In previous studies, in order to solve the problem of a large number of redundant alarms generated by IDS, some scholars proposed the hierarchical alarm clustering algorithm. With the development of the swarm intelligence optimization algorithm, the WOA has been widely used to solve various optimization problems. In this paper, we propose two new methods to solve the problem of alarm reduction by applying the WOA to hierarchical clustering, namely WOAHC-L and WOAHC-G. Through experiments on the UNSW-NB data set, we prove that WOAHC-L can solve the problems of the clustering falling into the local optimum and the poor clustering quality well. Experimental results show that WOAHC-L can achieve 95.2% clustering accuracy and reduce redundant alarms by 96.1%. Compared with WOAHC-L, which generates more than 11 clusters, WOAHC-G reduces the problem of clustering overlap by generating 8 accurate clusters.

Although our new method achieves good results in dealing with redundant alarms, it still has some limitations. For example, WOAHC-L can solve the problem of premature convergence in the clustering process well, but it is easy to produce an excessive number of repeated clusters. WOAHC-G solves this problem well, but it requires the space size of multiple orders of magnitude to encode the hierarchical tree. Meanwhile, WOAHC-G requires a prior number of clusters to ensure that the clustering is not repeated, which will make it difficult to deal with real network problems.

Future work will consider shifting the focus of work to the processing of security events in the real network environment and studying how to adjust a more excellent fitness function to obtain more accurate clustering. In addition, the adaptive setting of the clustering number of WOAHC-G will also be a key point of future work.

Author Contributions: All authors contributed to this manuscript. Conceptualization, L.W.; methodology, L.W.; data curation, L.W.; supervision L.G.; validation, L.G. and Y.T.; resources, L.G.; writing—original draft preparation, L.W.; writing—review and editing, L.G., Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://research.unsw.edu.au/projects/unsw-nb15-dataset, accessed on 18 November 2021].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, J.; Gu, L.; Chen, K. An Efficient Alert Aggregation Method Based on Conditional Rough Entropy and Knowledge Granularity. *Entropy* **2020**, *22*, 324. [CrossRef]
2. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.K.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A taxonomy of network threats and the effect of current datasets on intrusion detection systems. *IEEE Access* **2020**, *8*, 104650–104675. [CrossRef]
3. Masdari, M.; Khezri, H. A survey and taxonomy of the fuzzy signature-based Intrusion Detection Systems. *Appl. Soft Comput.* **2020**, *92*, 106301. [CrossRef]
4. Aldweesh, A.; Derhab, A.; Emam, A. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowl.-Based Syst.* **2019**, *189*, 105124. [CrossRef]
5. Siddique, K.; Akhtar, Z.; Khan, F.A.; Kim, Y. KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research. *Computer* **2019**, *52*, 41–51. [CrossRef]
6. Ingre, B.; Yadav, A. Performance analysis of NSL-KDD dataset using ANN. In Proceedings of the 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, 2–3 January 2015; pp. 92–96.
7. Alkasassbeh, M.; Al-Naymat, G.; Hassanat, A.B.; Almseidin, M. Detecting Distributed Denial of Service Attacks Using Data Mining Techniques. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 436–445. [CrossRef]
8. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6.
9. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. A detailed analysis of the cicids2017 data set. In Proceedings of the International Conference on Information Systems Security and Privacy, Funchal-Madeira, Portuga, 22–24 January 2018; Springer: Cham, Switzerland, 2018; pp. 172–188.
10. Damasevicius, R.; Venckauskas, A.; Grigaliunas, S.; Toldinas, J.; Morkevicius, N.; Aleliunas, T.; Smuikys, P. LITNET-2020: An Annotated Real-World Network Flow Dataset for Network Intrusion Detection. *Electronics* **2020**, *9*, 800. [CrossRef]
11. Dinh, D.T.; Huynh, V.N.; Sriboonchitta, S. Clustering mixed numerical and categorical data with missing values. *Inf. Sci.* **2021**, *571*, 418–442. [CrossRef]
12. Pattanodom, M.; Iam-On, N.; Boongoen, T. Clustering data with the presence of missing values by ensemble approach. In Proceedings of the 2016 Second Asian Conference on Defence Technology (acdt), Chiang Mai, Thailand, 21–23 January 2016; pp. 151–156.
13. Boluki, S.; Dadaneh, S.Z.; Qian, X.; Dougherty, E.R. Optimal clustering with missing values. *BMC Bioinform.* **2019**, *20*, 321. [CrossRef] [PubMed]
14. Ahmed, T.; Siraj, M.M.; Zainal, A.; Mat Din, M. A taxonomy on intrusion alert aggregation techniques. In Proceedings of the 2014 International Symposium on Biometrics and Security Technologies (ISBAST), Kuala Lumpur, Malaysia, 26–27 August 2014; pp. 244–249.
15. Husák, M.; Čermák, M.; Laštovička, M.; Vykopal, J. Exchanging security events: Which and how many alerts can we aggregate? In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8–12 May 2017; pp. 604–607.
16. Milan, H.S.; Singh, K. Reducing false alarms in intrusion detection systems—A survey. *Int. Res. J. Eng. Technol.* **2018**, *5*, 9–12.
17. Tian, W.; Zhang, G.; Liang, H. Alarm clustering analysis and ACO based multi-variable alarms thresholds optimization in chemical processes. *Process. Saf. Environ. Prot.* **2018**, *113*, 132–140. [CrossRef]
18. Hachmi, F.; Boujenfa, K.; Limam, M. Enhancing the Accuracy of Intrusion Detection Systems by Reducing the Rates of False Positives and False Negatives Through Multi-objective Optimization. *J. Netw. Syst. Manag.* **2018**, *27*, 93–120. [CrossRef]
19. Liu, L.; Xu, B.; Zhang, X.; Wu, X. An intrusion detection method for internet of things based on suppressed fuzzy clustering. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 113. [CrossRef]
20. Zhang, R.; Guo, T.; Liu, J. An IDS alerts aggregation algorithm based on rough set theory. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *322*, 062009. [CrossRef]
21. Li, W.; Meng, W.; Su, C.; Kwok, L.F. Towards False Alarm Reduction Using Fuzzy If-Then Rules for Medical Cyber Physical Systems. *IEEE Access* **2018**, *6*, 6530–6539. [CrossRef]
22. Hu, Q.; Lv, S.; Shi, Z.; Sun, L.; Xiao, L. Defense against advanced persistent threats with expert system for internet of things. In Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications; Guilin, China, 19–21 June 2017; Springer: Cham, Switzerland, 2017; pp. 326–337.
23. Zaeri-Amirani, M.; Afghah, F.; Mousavi, S. A feature selection method based on shapley value to false alarm reduction in icus a genetic-algorithm approach. In Proceedings of the 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 18–21 July 2018; pp. 319–323.
24. Sabino, S.; Grilo, A. Routing for Efficient Alarm Aggregation in Smart Grids: A Genetic Algorithm Approach. *Procedia Comput. Sci.* **2018**, *130*, 164–171. [CrossRef]
25. Mannani, Z.; Izadi, I.; Ghadiri, N. Preprocessing of Alarm Data for Data Mining. *Ind. Eng. Chem. Res.* **2019**, *58*, 11261–11274. [CrossRef]

26. Hashim, S.H. Intrusion detection system based on data mining techniques to reduce false alarm rate. *Eng. Technol. J.* **2018**, *36 (Pt B)*, 110–119.
27. Mavrovouniotis, M.; Li, C.; Yang, S. A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm Evol. Comput.* **2017**, *33*, 1–17. [CrossRef]
28. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
29. Gharehchopogh, F.S.; Gholizadeh, H. A comprehensive survey: WOA and its applications. *Swarm Evol. Comput.* **2019**, *48*, 1–24. [CrossRef]
30. Wang, Y.; Meng, W.; Li, W.; Liu, Z.; Liu, Y.; Xue, H. Adaptive machine learning based alarm reduction via edge computing for distributed intrusion detection systems. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e5101. [CrossRef]
31. Toldinas, J.; Venčkauskas, A.; Damaševičius, R.; Grigaliūnas, Š.; Morkevičius, N.; Baranauskas, E. A Novel Approach for Network Intrusion Detection Using Multistage Deep Learning Image Recognition. *Electronics* **2021**, *10*, 1854. [CrossRef]
32. Weiß, I.; Kinghorst, J.; Kröger, T.; Pirehgalin, M.F.; Vogel-Heuser, B. Alarm flood analysis by hierarchical clustering of the probabilistic dependency between alarms. In Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), Porto, Portugal, 18–20 July 2018; pp. 227–232.
33. Fahimipirehgalin, M.; Weiss, I.; Vogel-Heuser, B. Causal inference in industrial alarm data by timely clustered alarms and transfer entropy. In Proceedings of the 2020 European Control Conference (ECC), St. Petersburg, Russia, 12–15 May 2020; pp. 2056–2061.
34. Alharbi, A.; Alosaimi, W.; Alyami, H.; Rauf, H.; Damaševičius, R. Botnet Attack Detection Using Local Global Best Bat Algorithm for Industrial Internet of Things. *Electronics* **2021**, *10*, 1341. [CrossRef]
35. Abu Khurma, R.; Almomani, I.; Aljarah, I. IoT Botnet Detection Using Salp Swarm and Ant Lion Hybrid Optimization Model. *Symmetry* **2021**, *13*, 1377. [CrossRef]
36. Zhang, J.; Yu, B.; Li, J. Research on IDS Alert Aggregation Based on Improved Quantum-behaved Particle Swarm Optimization. In Proceedings of the Computer Science and Technology(CST2016), Shenzhen, China, 8–10 January 2016; pp. 293–299.
37. Lin, H.C.; Wang, P.; Lin, W.H.; Chao, K.M.; Yang, Z.Y. Identifying the Attack Sources of Botnets for a Renewable Energy Management System by Using a Revised Locust Swarm Optimisation Scheme. *Symmetry* **2021**, *13*, 1295. [CrossRef]
38. Ibrahim, N.M.; Zainal, A. A Feature Selection Technique for Cloud IDS Using Ant Colony Optimization and Decision Tree. *Adv. Sci. Lett.* **2017**, *23*, 9163–9169. [CrossRef]
39. Osanaiye, O.; Cai, H.; Choo, K.-K.R.; Dehghantanha, A.; Xu, Z.; Dlodlo, M. Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. *EURASIP J. Wirel. Commun. Netw.* **2016**, *2016*, 130. [CrossRef]
40. Lu, X.; Du, X.; Wang, W. An Alert Aggregation Algorithm Based on K-means and Genetic Algorithm. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *435*, 012031. [CrossRef]
41. Yang, X.S.; Gandomi, A.H. Bat algorithm: A novel approach for global engineering optimization. *Eng. Comput.* **2012**, *29*, 464–483. [CrossRef]
42. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
43. Yapici, H.; Cetinkaya, N. A new meta-heuristic optimizer: Pathfinder algorithm. *Appl. Soft Comput.* **2019**, *78*, 545–568. [CrossRef]
44. Julisch, K. Clustering intrusion detection alarms to support root cause analysis. *ACM Trans. Inf. Syst. Secur.* **2003**, *6*, 443–471. [CrossRef]
45. Julisch, K. Using Root Cause Analysis to Handle Intrusion Detection Alarms. Ph.D. Thesis, University of Dortmund, North Rhine-Westphalia, Germany, 2003.
46. Wang, J.; Wang, H.; Zhao, G. A GA-based Solution to an NP-hard Problem of Clustering Security Events. In Proceedings of the 2006 International Conference on Communications, Circuits and Systems, Guilin, China, 25–28 June 2006; pp. 2093–2097.
47. Wang, J.; Xia, Y.; Wang, H. Minining Intrusion Detection Alarms with an SA-based Clustering Approach. In Proceedings of the 2007 International Conference on Communications, Circuits and Systems, Kokura, Japan, 11–13 July 2007; pp. 905–909.
48. Mafarja, M.; Mirjalili, S. Whale optimization approaches for wrapper feature selection. *Appl. Soft Comput.* **2018**, *62*, 441–453. [CrossRef]
49. Frank, E.; Hall, M.; Trigg, L.; Holmes, G.; Witten, I. Data mining in bioinformatics using Weka. *Bioinformatics* **2004**, *20*, 2479–2481. [CrossRef] [PubMed]

symmetry

MDPI

Article

Effective Intrusion Detection System to Secure Data in Cloud Using Machine Learning

Ammar Aldallal * and Faisal Alisa

Telecommunication Engineering Department, College of Engineering, Ahlia University, Manama 10878, Bahrain; faisal.alisa@khuh.org.bh
* Correspondence: aaldallal@ahlia.edu.bh

Abstract: When adopting cloud computing, cybersecurity needs to be applied to detect and protect against malicious intruders to improve the organization's capability against cyberattacks. Having network intrusion detection with zero false alarm is a challenge. This is due to the asymmetry between informative features and irrelevant and redundant features of the dataset. In this work, a novel machine learning based hybrid intrusion detection system is proposed. It combined support vector machine (SVM) and genetic algorithm (GA) methodologies with an innovative fitness function developed to evaluate system accuracy. This system was examined using the CICIDS2017 dataset, which contains normal and most up-to-date common attacks. Both algorithms, GA and SVM, were executed in parallel to achieve two optimal objectives simultaneously: obtaining the best subset of features with maximum accuracy. In this scenario, an SVM was employed using different values of hyperparameters of the kernel function, gamma, and degree. The results were benchmarked with KDD CUP 99 and NSL-KDD. The results showed that the proposed model remarkably outperformed these benchmarks by up to 5.74%. This system will be effective in cloud computing, as it is expected to provide a high level of symmetry between information security and detection of attacks and malicious intrusion.

Keywords: intrusion detection system; genetic algorithm; support vector machine; machine learning; fitness function

Citation: Aldallal, A.; Alisa, F. Effective Intrusion Detection System to Secure Data in Cloud Using Machine Learning. *Symmetry* **2021**, *13*, 2306. https://doi.org/10.3390/sym13122306

Academic Editors: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Received: 10 October 2021
Accepted: 16 November 2021
Published: 3 December 2021

1. Introduction

Cloud computing is one of the latest service innovations in the field of IT. The primary advantage of cloud computing is that it enables access without constraints of location and time. Cloud computing supports mobile and collaborative applications/services, enables the flexibility of controlling storage capacities, and provides lower costs. Moreover, cloud services are multisource, permitting the end-users to use multiple service providers based on their requirements. The use of cloud computing also reduces capital expenditures, power usage, and physical space and maintenance requirements for on-site storage.

As cloud computing services become more and more common, a large number of companies, banks, and governments have adopted this technology. This transition also exposed these systems to many kinds of cyberattacks by hackers and intruders, warranting robust security mechanisms. Many cloud service companies provide several security services as applications. An example is the Amazon Web Services (AWS) store, which provides services with limited validity and dates depending on the period of service license.

With increasing volumes of data, particularly important medical records, the need for continuous backup and updates is evident. Healthcare data is confidential and is an attractive target for hackers to manipulate or use for illegitimate purposes such as financial gain or political motives [1]. Health data and medical records can include specific patient history, information on prescriptive drugs and medical devices, and other confidential patient information. The privacy of this information is of primary importance. Therefore, online attacks including identity theft should be avoided, as they lead to illegitimate access to financial, social, and banking records [1].

When adopting cloud computing technologies, cybersecurity must be a priority aspect for healthcare services because of the confidentiality of patient and operational data. Existing intrusion detection systems (IDS) operate on the mechanism of signature or anomaly detection. When the detection mechanism is not suitable, patient and operational data may be stolen. Cybersecurity strategy helps to detect and protect against malicious actors while helping to improve an organization's defense against cyberattacks.

Existing studies in international hospitals have been limited to user awareness of the importance of cybersecurity such as the use of strong passwords, deletion or filtration of unwanted emails, data encryption, confidential treatment of credentials, careful access of information, and swift reporting of any security breaches [2]. Intrusion detection systems are also in place for health sectors, but their local infrastructure may not be free from vulnerabilities. In contrast, the cloud provides a significantly useful and secure service to manage the operations within hospitals or any other organization.

Sooner or later, all government IT operations are bound to be run on the cloud. Currently, the Ministry of Health in Bahrain is switching its systems over to the cloud environment [3]. The cloud provides specific types of security depending on the services provided by the cloud service provider. In certain cases, highly confidential data cannot be hosted on the cloud, while some operational data can be stored online. The data managed on in-house servers must be protected at the same level of security when compared with cloud security. This requires significant work to optimize cloud resource security, which helps to build trust for deploying confidential and sensitive medical data on the cloud.

Several studies have been conducted on hybrid IDS applying different techniques, some of which have proven to have high efficiency while others resulted in only acceptable efficiency. In this research, the implementation of the hybrid IDS was developed by applying both improved genetic algorithm (GA) and support vector machine (SVM) techniques to achieve the best possible results. The proposed system can be used to secure health data in the cloud.

The dataset used in evaluating any IDS has a crucial role in identifying its effectiveness. Therefore, a vast number of research studies have adopted the KDD CUP 99 dataset to evaluate their systems. In this regard, a few examples include [4–7]. Another well-known dataset used in this field is NSL-KDD, which was examined by [8–11]. These datasets are relatively old and have a limited number of features, making them unreliable in simulating current systems and environments.

This work is intended to be applied to cloud data for Bahrain hospitals, but because of the difficulty in accessing their data to develop the intended system, a predefined dataset that mimics the existing cloud data of hospitals, CICIDS2017, was used instead. CICIDS2017 is one of the most recent datasets applied in machine learning applications. The present study is significant because it aimed to:

1. Identify the set of optimal features in the new dataset CICIDS2017, which contains normal and most up-to-date common attacks. This was done to find a set of selected features that improved the performance of the detection mechanism.
2. Examine the effect of newly developed fitness functions on identifying intrusions. The present study aimed to do this using a multifactor fitness function to measure the performance of IDS.
3. Determine the best combination of parameter values for an SVM to produce the maximum detection rate of intrusions within the CICIDS2017 dataset.
4. Conduct a comparative study between the CICIDS2017 dataset and the popular KDD CUP 99 dataset. The outcome was used to generalize the efficiency of the proposed system.

This system will be effective in cloud computing, as it is expected to provide a high level of information security, which will protect data in the cloud environment against several types of malicious attacks.

The rest of this paper is organized as follows: Section 2 elaborates on the background about IDS and related studies using GA and SVM. The proposed hybrid IDS is presented in Section 3, and the proposed improved GA and SVM are explained thoroughly. Explanation of the experiments and data processing along with the results are delineated in Section 4. Finally, the conclusion and future work are the contents of Section 5.

2. Background and Literature Review

Intrusion detection systems (IDS) constitute a vital topic studied extensively by researchers. The techniques applied to IDS are basically categorized into anomaly-based IDS, signature-based IDS, and hybrid techniques. When manipulating huge amounts of data related to IDS, the first step is dataset preprocessing. The main stage of preprocessing is feature selection. Once features are selected, machine learning algorithms are applied to classify the normal and abnormal behavior of intruders. In the following text, we highlight the categories of IDS, followed by feature selection techniques, and afterwards, we introduce the hybrid machine learning techniques used in the IDS in the present study.

2.1. Concepts of IDS and Related Work

An IDS is a system that monitors all outgoing and incoming requests in cloud computing to detect any abnormal or normal activity and is a method of detecting anomalies and misuse intrusion.

Infiltration and intrusions are the main problems in the network and cloud computing, as all services are provided over the internet and are susceptible to cyberattacks. IDS software should be scalable, dynamic, and self-adaptive, with high efficiency to use. The intrusion detection process is divided into three steps [12]:

- collecting data from the monitored systems;
- analyzing the collected data to classify it as normal or intrusion;
- alerting the admin about the possibility of intrusion to the concerned system so that they can take necessary action to prevent intrusion and protect the system.

2.1.1. Anomaly-Based IDS

Anomaly-based IDS detect misuse in the cloud environment or penetrations and classify them into normal and abnormal user behavior through a system that collects all the information on normal user behavior or action over a period of time. Then, a statistical test is performed to check whether a suspected behavior is linked to a normal user's behavior or not. Figure 1 presents the general architecture of anomaly-based IDS. The difficulty in maintaining this type of IDS is that it cannot be updated without losing the data that the previous system was trained on. Also, the accuracy of identification is low, which gives a high number of false positives for this type of system. Anomaly-based IDS has been addressed by several researchers as follows.

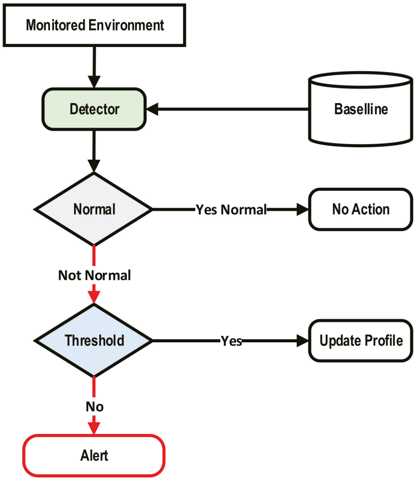


Figure 1. Anomaly-based IDS architecture [13].

Aljawarneh et al. [14] proposed an IDS based on anomaly detection using feature selection, which resulted in an efficient hybrid model. This new hybrid model was useful in estimating intrusions based on network activity and transaction of optimal data options that were available for training. More optimizing techniques were required to create a better IDS model that has a better accuracy rate.

In reference [15], a web application profile was developed by learning constants (e.g., to log in, the value for the user should be the same as the value when logging in). To verify whether the constants were violated, the source code was used. If a static element violation was observed, an anomaly was entered.

In reference [16], an anomaly-based IDS was developed by analyzing the workflow of all web sessions. It consisted of small business groups of data objects. These groups were used and applied to typical data access sequences for the workflow processes. The authors used a hidden Markov model (HMM) application. The results showed that this approach could detect anomalous web sessions and lent evidence to suggest that the clustering approach could achieve relatively low FPR while maintaining its detection accuracy.

Le et al. [17] developed the double-guard framework, which tested both database and web server logs to detect attacks that leaked information. They reported a 0.6% FP rate for dynamic webpages and a 0% FP rate for static webpages.

Nascimento and Correia [18] studied an IDS in which they collected the dataset from a scale web app and trained the dataset. They considered "GET" requests and did not consider "POST" types of requests or response pages. They took logs from the T-Shark tool and converted them to a common format. The filtered data was created by accessing the subapplications. They used nine detection models.

Ariu [19] developed a host-based IDS to protect web applications from attacks by using HMMs. This system was used to model a series of attributes and values received by a web app. To calculate the different parameters and values, they used several HMMs and combined them to arrive at a specific demand on the probability that was generated from the training dataset.

In reference [20], a firewall was developed as a web application to detect any anomalous request and capture the behavior through an "XML" file that defined the required attributes of parameter values. Input values that deviated from the profile were considered attacks. However, this approach produced *FP* warnings because it did not consider the path information and page to be more accurate.

2.1.2. Signature-Based IDS

In signature-based IDS, known threats previously discovered and identified from the system help to identify threats in the future. This technique is used to compare the incoming network pattern. An intrusion is detected when the incoming network pattern matches the signature. The advantage of this type of detection method is that it is easy to develop and understand by knowing the network behavior signatures. This technique has very high accuracy and a minimum number of false positives in detecting known attacks. Also, it can add new signatures into the database without changing or modifying existing ones.

The main drawback issue of this IDS technique is that these types of intrusion detecting systems are not able to detect new types of attacks or unknown attacks; a slight change in the pattern or slight variation can fool it. Figure 2 presents the general architecture of signature-based IDS. A few examples of related work that applied signature-based IDS are listed here.

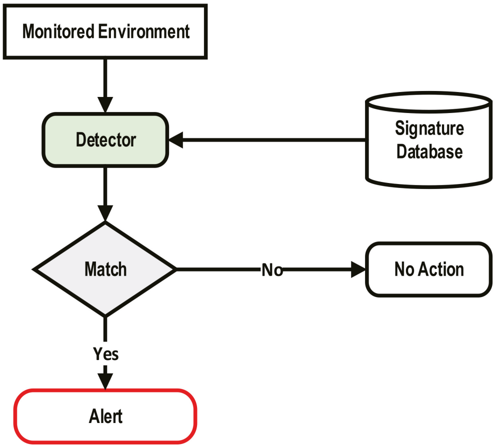


Figure 2. Signature-based IDS architecture [13].

Saraniya [21] developed a network intrusion detection system (NIDS) to secure the network using a signature-based IDS algorithm. It succeeded in capturing packets sent across the entire network using mixed mode and comparing traffic to designer attack signatures. It secured the network and reduced the memory space in the environment. Signature-based IDS are not able to detect emerging and unknown attacks because the signature database must be manually reviewed for every new type of intrusion discovered.

Gao and Morris [22] presented a study on cyberattacks and signature-based IDS for MODBUS based industrial control systems. The rules described were designed to detect the attacks described earlier. The rules were divided into two types: independent and state-based rules. Independent rules were those that analyzed one MODBUS packet, looking for a specific signature match. Standalone rules were those that were enforced with the Snort intrusion detection tool.

Uddin et al. [23] proposed a signature-based distributed IDS using a mobile agent, to transfer the signature database from a large complementary database to a small signature database, then update the databases regularly as new signatures are discovered. The results of the proposed model indicated that IDS worked better than regular systems that used only one database of chain signatures.

Kumar and Gobil [24] implemented a signature-based IDS. They used three tools to develop this IDS system: SNORT, BASE, and TCP REPLAY. This system could detect and analyze intrusions in the network traffic in real time.

The main characteristics and limitations of anomaly-based, signature-based, and hybrid IDS are listed in Table 1.

Table 1. Summary of intrusion detection techniques [17].

Anomaly-Based IDS	
Characteristics	**Limitations**
This technique can detect unknown attacks. Low rate of false alarm for unknown attacks. For IDS, collects the behavior of the network for statistics on testing and performance.	This technique is time-consuming. It needs many features to detect attacks correctly.
Signature-Based IDS	
Characteristics	**Limitations**
The detection rate of this technique is very high for known previous attacks. This technique detects attacks by matching detected packet patterns with previously acquired patterns.	Very high rate of false alarm for unknown attacks. Cannot detect a variant of a known attack or new attacks.
Hybrid Techniques for IDS	
Characteristic	**Limitation**
It is a combination of two or more techniques. Detection accuracy is very high.	Computational cost is very high.

2.2. Feature Selection

The continuous increase in the volume of data makes real-time intrusion detection a difficult task. Feature selection (FS) is a technique used to reduce computation complexity by selecting the relevant features and eliminating many redundant and needless features from the data to produce effective learning model [25,26].

When dimensionality of a feature is high, the selection of meaningful features becomes a difficult challenge. Bioinspired metaheuristic algorithms are efficient in addressing such challenges. They provide high-quality solutions in a reasonable time and within affordable resources. Moreover, they decrease the computational complexity of classification and improve its accuracy [27].

Intelligent IDS work by collecting and analyzing large amounts of data from several areas. These data contain various redundant and irrelevant features, which results in increased overfitting and processing time and reduced detection rate. Therefore, feature selection needs to be applied as a preprocessing step to improve system performance and accuracy while reducing the dataset size [28].

Different approaches have been developed by researchers to reduce the number of features while implementing IDS. In reference [29], a genetic algorithm was applied to select the optimal set of features from KDD CUP 99 and UNSW-NB15. The authors applied a new fitness function, which was composed of TPR, FPR, and the number of selected features, with specific weights given for each factor. The selected features were then passed to the classification stage, at which SVM was applied. The number of optimized features ranged between 7 and 10 for different types of attacks. The maximum accuracy of classification of normal traffic was 99.05%, while the accuracy of classification ranged from 98.25% for R2L to 100% for U2R.

Parimala and Kayalvizhi [30] developed a two-stage feature selection as a prestage for an IDS that provided secure communication in a wireless environment. The first stage applied a conditional random field (CRF) for the initial selection of features. In the second stage, spider monkey optimization (SMO) was applied to the selected feature to identify the most useful features from the dataset. This model was applied to the NSL-KDD dataset, which is a wireless dataset that has 41 features. This model extracted 16 features. The authors did not provide exact figures of achieved accuracy, but the results outperformed four existing works according to their experiments.

By using the same dataset, NSL-KDD, Reference [31] examined the performance of four feature selection methods: information gain, gain ratio, chi squared, and relief. Their performance was analyzed by applying them on four machine learning algorithms: J48, random forest, naïve Bayes, and k-nearest neighbor (KNN). The obtained results showed that feature selection greatly improved the performance of the IDS but had a slightly negative effect on accuracy.

In addition to information gain and chi squared, Reference [32] applied correlation-based feature selection. These selection methods were applied to the NSL-KDD dataset. The decision tree classifier was the ML algorithm adopted in their IDS model. Initially, the dataset consisted of 41 features. During the preprocessing stage, the nonnumeric data was converted to numeric data, producing a total of 122 features. Their proposed methods selected 20 optimal features out of these 122. The maximum accuracy obtained was 81%.

Reference [33] developed a cloud IDS (CIDS) that started by feature selection. They used an efficient correlation-based feature selection approach, which extended correlation-based feature selection and mutual information feature selection. The feature selection process consisted of two stages. The first stage was used to recognize important features, so it reduced the dimensionality of the dataset. This was achieved by reducing the pairwise calculations of the correlation between features. To distinguish all different classes, this method was combined with the LIBSVM classifier. In the second stage, features were selected such that they were highly relevant to a given class c but not relevant to the selected features. This approach was applied to both the KDD CUP 99 and NSL-KDD datasets. The proposed method enhanced the time required to obtain the optimum set of features and reduced the number of selected features to 10.

Reference [26] proposed a hybrid IDS that consisted of feature selection, density-based spatial clustering of application with noise (DBSCAN), K-Mean++ clustering, and SMO classification. The feature selection adopted to be applied on the KDDCUP 99 dataset was a genetic algorithm that selected 13 features for further processing. The average accuracy achieved in this model was 96.92%.

The feature selection model implemented by [28] consisted of two parts. The first part was the attribute evaluator, which evaluated each attribute separately. The second part was the search method, in which different combinations were tried to obtain the selected list of features. This model was applied to the NSL-KDD dataset. Feature selection was performed on three classes: all attack types (23 types), main attack types (5 types), and two attack types (normal and abnormal). This approach was then examined using three classification models: random forest, J48, and naïve Bayes. The results obtained for the two-attacks type ranged from 98.69 to 99.41%.

2.3. Hybrid Intrusion Detection System Based on Machine Learning

Researchers have implemented IDS using several ML techniques, among other techniques. To focus, the discussion in this section is limited to two ML techniques: the genetic algorithm (GA) and support vector machine (SVM). Before presenting related work conducted using GA and SVM, it is worth shedding some light on these techniques.

2.3.1. Genetic Algorithm Overview

Genetic algorithms (GA) constitute a family of mathematical models that operate on the principles of selection and natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms work by transforming selected problems using chromosome-like data and developing chromosomes using selection factors, recombination, and mutations.

The genetic algorithm process usually begins with a random set of chromosomes. These chromosomes are representations of a problem that must be solved. Depending on the attributes of a problem, the positions of each chromosome are encoded in numbers, letters, or bits. These positions are referred to as genes and are randomly changed within a range during development. The set of chromosomes during the evolution stage is called a population. The evaluation function is used to calculate the GOODNESS of each chromosome. There are two primary factors during the evaluation. Crossover is used to simulate the transformation and the natural reproduction of the species. Chromosomal selection for survival and synthesis is biased towards the fittest chromosomes. Genetic algorithms can model virtually any type of constraint in the form of partial functions or by using different chromosome coding schemes specifically designed for a specific problem. Figure 3 shows the structure of a simple genetic algorithm.

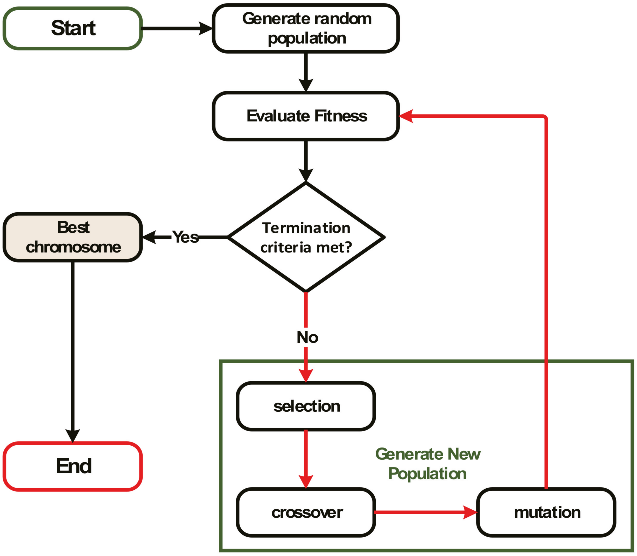


Figure 3. Structure of a genetic algorithm.

Since GAs are best used for optimization, they can be used to generate the best rules for a network connection that can be used to classify the network behavior as intrusive or normal. In addition, GAs can be used to select the optimal set of features from a dataset that can be used in the process of intrusion detection. They can also be used in computer security applications to find optimal solutions to specific problems.

2.3.2. SVM-Based IDS and Related Work

The support vector machine is a machine learning technique known as the best learning algorithm for classification. It is one of the most popular supervised ML algorithms [4,34]. The SVM is a type of pattern classification and regression with a variety of kernel functions. It has been applied to several pattern recognition applications. SVMs have been used mainly for binary classification. The idea is to find a line or hyperplane that separates two classes such that it is as far as possible from the closest instances of each class. Instances of different groups are separated by an area called the margin. The closest instances to the hyperplane are called support vectors. It is required to have the margin as big as possible; this is important to enhance the efficiency of the classification of newly added instances. The boundary function of the largest margin can be computed as follows [35]:

$$Minimize\ W\ (\alpha)\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} X_I X_J \alpha_I \alpha_J k\ (y_i,\ y_j) - \sum_{i=1}^{N} \alpha_i$$

where α is a vector of N variables, C is the soft margin parameter, $C > 0$, and $k\ (y_i,\ y_j)$ represents the kernel function of SVMs. Kernel functions are used in SVMs to classify instances of the dataset into different categories. Four main types of kernel functions are listed here [35]:

1. linear kernel: $k\ (y_i, y_j) = y_i^T \cdot y_j$;
2. polynomial kernel: $k\ (y_i, y_j) = (\gamma\ y_i^T \cdot y_j + r)^d$, $\gamma > 0$;
3. radial basis function (RBF) kernel: $k\ (y_i, y_j) = exp(-\gamma \mid\mid y_i, y_j{}^2 \mid\mid)$, $\gamma > 0$;
4. sigmoid kernel: $y_i, y_j = tanh\ (\gamma\ y_i^T \cdot y_j + r)$

where γ, r and d are kernel parameters. The concepts of the margin, separation hyperplane, and support vector are illustrated in Figure 4.

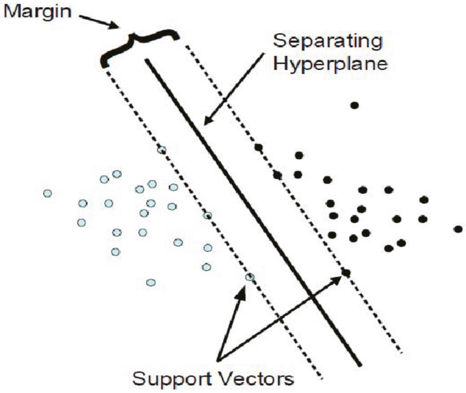


Figure 4. Classical example of SVM linear classifier [4].

This algorithm has been applied in information security to detect intrusion. The main advantage of using a support vector machine for an IDS is its speed as a capability of detecting intrusion in real time. SVMs have become a popular technique for detecting anomalous intrusion because of their good circular nature and their ability to overcome the curse of dimensionality. They are also useful for finding global minimum actual risk using structural risk reduction, because they can generalize well with kernel tricks even in high-dimensional spaces under small-training-sample conditions.

The support vector machine can determine the appropriate setting parameters because it does not depend on traditional experimental risks such as neural networks and can learn a wide range of patterns that can expand. Support vector machines can also dynamically update training patterns whenever there is a new pattern during classification. Several researchers have applied SVMs as part of intrusion detection, as discussed below.

2.3.3. Related Work on Existing Hybrid IDS Using GA and SVM

A set of research works have adopted GA as a base for IDS, while others have adopted SVM. GA has been applied in IDS by several researchers, as follows. Neha Rai and Khushbu Rai [36] proposed an intrusion detection system using genetic algorithm techniques. To create a new population, they used rank selection for selection, single point for crossover, and bitwise for mutation. Reference [34] proposed an intrusion alarm based on a genetic algorithm and support vector machine. In the GA part, an accuracy fitness function was applied, and to create a new population, tournament selection in the selection part, two-point crossover, and simple mutation in the reproduction part were used. Ojugo et al. [37] proposed a genetic algorithm rule-based IDS. In this system, they applied the confidence fitness function, and to create a new population, they used tournament selection and two-point crossover. Bhattacharjee et al. [38] proposed an IDS using a vectorized fitness function in a genetic algorithm and used uniform crossover. Prakash et al. [39] built an effective IDS using GA-based feature selection. They applied an accuracy fitness function, and to create a new population, they used tournament selection and uniform crossover. Reference [40] proposed a GA-based approach to NIDS. In this system, a confidence fitness function was applied, and to create a new population, two-point crossover was used. Pawar and Bichkar [41] proposed a NIDS using GA with VLC; to create a new population, they used single-point crossover.

Moukhafi et al. [4] proposed a new method of IDS that used an SVM optimized by a GA to improve the efficiency of detecting unknown and known attacks. They used the particle swarm optimization algorithm to select influential features. The detection rate (DR) achieved was 96.38% when applied to 10% of the KDD CUP 99 dataset. The small size of the dataset facilitated the process of the GA–SVM classifier.

Another hybrid IDS was developed in [5]. It used kernel principal component analysis (KPCA), an SVM, and a GA. The kernel function of the SVM was RBF modified to include additional two parameters to obtain better feature values. These parameters were the mean value and the mean square difference values of feature attributes. The chromosome of the GA had three representations. The KDD CUP 99 dataset was used. The DR produced was 95.26%.

Al-Yaseen et al. [6] developed a hybrid IDS which uses K-means, SVM, and ELM. The focus in this work lay on building a training set with high quality that required little time for classification by using K-means. The KDD CUP 99 dataset was used. The DR produced was 95.17%.

Feng et al. [7] developed a hybrid IDS that used an SVM and an ant colony network. The hybrid system benefited from the high classification rate and runtime efficiency produced from the combination of these two algorithms. They used the KDD CUP 99 dataset and managed to achieve a DR of 94.86%.

Tao et al. [34] proposed an improved intrusion detection algorithm based on the GA and SVM methods. This paper proposed an alarm intrusion detection algorithm. Feature selection and weight and parameter optimization of the support vector machine were based on the genetic algorithm. The simulations and experimental results showed that the intrusion detection technology based on the "GA and SVM" proposed increased the intrusion DR, the Accuracy Rate, and the TPR while reducing the FPR. SVM training time was also reduced.

Agarwal and Mittal [42] proposed a hybrid approach for the detection of anomaly network traffic using data mining techniques. In this hybrid model, they applied entropy of network and SVM techniques. This hybrid method worked well and gave high accuracy for the detection of attack traffic with few false alarms. This method was not dynamic and did not detect or decide whether there was an attack.

Serpen and Aghaei [43] proposed a host-based misuse IDS using PCA feature extraction and a KNN classification algorithm. This system used basic analysis Eigen traces to extract data of the OS for a "trace data and k-nearest neighbor" algorithm for classification. This design exhibited very high performance and the ability to detect intrusion and attack plus the type of intrusion. This system worked on the Linux OS only.

Praveen et al. [44] proposed a hybrid IDS algorithm for private cloud services. They applied anomaly intrusion and misuse intrusion, and they used the NET framework as a front end and a SQL server as a back end to implement the algorithm. They conducted an overall study to build a hybrid IDS that would help to detect all types of intrusion in the cloud environment. According to the authors, the major characteristics that a hybrid IDS must have are a dynamic nature, self-adaptiveness, and scalability by the OS in the network and host.

3. The Proposed Hybrid IDS

The proposed hybrid IDS is a combination of two machine learning techniques, the GA and SVM, in which the GA is used for feature selection, which is considered as an optimization problem. Since the used dataset consisted of 79 features, and some of these featured had little to no effect on the identification of intrusion, a GA is applied to select the optimum number of features to maintain the high performance of IDS while reducing the overhead of the classification process. On the other hand, an SVM is applied to perform the actual classification of network data into normal and abnormal behaviors. However, before applying the algorithm to the dataset, it needs to be preprocessed so that the algorithm works smoothly on clean and consistent data. The preprocessing is explained in the following subsection.

3.1. Data Preparation

As mentioned in section one, one of the objectives of this work was to examine the proposed IDS on a recent dataset rather than the commonly used old dataset, KDD CUP 99. The dataset used in this work was the Intrusion Detection Evaluation Dataset [45]. Data covering only 5 days between 3 July 2017 and 7 July 2017 were used in this study. This data contained benign traffic and a limited number of several types of attacks such as brute-force FTP, brute-force SSH, SQL injection attack, and cross-site scripting. The dataset was known as CICIDS2017 and consisted of 170 K records and 79 features.

The first stage of preparation was to clean the data. Cleaning was done in terms of unifying the data type to integers, since some numerical values were entered as "infinity". Second, the missing values were replaced with zeros. Third, two columns were eliminated, as they were dramatically corrupted, causing the algorithm to fail. These columns were Fwd_Packets_s and Bwd_Packets_s. Fourth, the data was scaled, as some columns had one-figure values, while others had up to six-figure values. Scaling was performed using the sklearn function StandardScaler, which normalizes data according to the formula:

$$\frac{z = x - mean}{standard\ deviation}$$

The remaining features are listed in Table 2.

Table 2. Features of the CICIDS2017 dataset [45].

No.	Feature Name	No.	Feature Name	No.	Feature Name
1	Destination_Port	27	Bwd_IAT_Max	53	Avg_Bv/d_Segment_Size
2	Flow_Duration	28	Bwd_IAT_Min	54	Fwd_Header_Length
3	Total_Fwd_Packets	29	Fwd_PSH_Flags	55	Fwd_Avg_Bytes_Bulk
4	Total_Backward_Packets	30	Bwd_PSH_Flags	56	Fwd_Avg_Packets_Bulk
5	Total_Length_of_Fwd_Packets	31	Fv/d_URG_Flags	57	Fwd_Avg_Bulk_Rate
6	Total_Length_of_Bwd_Packets	32	Bwd_URG_Flags	58	Bwd_Avg_Bytes_Bulk
7	Fwd_Packet_Length_Max	33	Fwd_Header_Length	59	Bwd_Avg_Packets_Bulk
8	Fwd_Packet_Length_Min	34	Bwd_Header_Length	60	Bv/d_Avg_Bulk_Rate
9	Fwd_Packet_Length_Mean	35	Fv/d_Packets_s	61	Subflow_Fwd_Packets
10	Fwd_Packet_Length_Std	36	Bwd_Packets_s	62	Subflow_Fv/d_Bytes
11	Bwd_Packet_Length_Max	37	Min_Packet_Length	63	Subflow_Bv/d_Packets
12	Bwd_Packet_Length_Min	38	Max_Packet_Length	64	Subflow_Bv/d_Bytes
13	Bwd_Packet_Length_Mean	39	Packet_Length_Mean	65	Init_Win_bytes_forward
14	Bwd_Packet_Length_Std	40	Packet_Length_Std	66	Init_Win_bytes_backward
15	Flow_IAT_Mean	41	Packet_Length_Variance	67	act_data_pkt_fwd
16	Flow_IAT_Std	42	FIN_Flag_Count	68	min_seg_size_forward

Table 2. *Cont.*

No.	Feature Name	No.	Feature Name	No.	Feature Name
17	Flow_IAT_Max	43	SYN_Flag_Count	69	Active_Mean
18	Flow_IAT_Min	44	RST_Flag_Count	70	Active_Std
19	Fwd_IAT_Total	45	PSH_Flag_Count	71	Active_Max
20	Fwd_IAT_Mean	46	ACK_Flag_Count	72	Active_Min
21	Fwd_lATStd	47	URG_Flag_Count	73	Idle_Mean
22	Fwd_IAT_Max	48	CWE_Flag_Count	74	Idle_Std
23	Fwd_IAT_Min	49	ECE_Flag_Count	75	Idle_Max
24	Bwd_1_AT_Total	50	Down_Up_Ratio	76	Idle_Min
25	Bwd_IAT_Mean	51	Average_Packet_Size	77	Label
26	Bwd_lATStd	52	Avg_Fv/d_Segment_Size		

The second dataset used in this work was KDD CUP 99, which was used to benchmark the performance of the proposed system. The dataset consisted of only 42 features, which are listed in Table 3.

Table 3. Features of the KDD CUP 99 dataset [46].

No.	Feature Name	No.	Feature Name
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	Inum_compromised	34	dst_host_same_srv_rate
14	Iroot_shell	35	dst_host_diff_srv_rate
15	Isu_attempted	36	dst_host_same_src_port_rate
16	Inum_root	37	dst_host_srv_diff_host_rate
17	Inum_file_creations	38	dst_host_serror_rate
18	Inum_shells	39	dst_host_srv_serror_rate
19	Inum_access_files	40	dst_host_rerror_rate
20	Inum_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login	42	label

3.2. Genetic Algorithm (GA)

As aforementioned, GAs constitute a family of mathematical models that operate on the principles of selection and natural evolution. GAs have multiple parameters, each of which can be implemented in several methods. GAs represent one of the best techniques for optimization problems and feature selection. The following is a description of GA operators: initial population creation, crossover, mutation, and the proposed fitness function, which was used in the feature selection stage. These operators are summarized in Algorithm 1.

Algorithm 1: GA process

```
1: Create randomly first generation.
2: Evaluate fitness of each chromosome in the first generation.
3: Loop
4:     Create next generation following these steps:
5:         selection;
6:         crossover;
7:         mutation.
8: Until stop criteria is met.
9: The result is the best chromosome's content of the last generation.
```

3.2.1. Initial Population

The dataset utilized in this work consisted of 78 features, which was a relatively high number and could affect the quality of the output. Therefore, among these 78, the GA chose 25 features randomly to start the initial population. Knowing that the GA was tested with different numbers of features, from 10 to 25, the results were close, and the training and testing times of the data were adequate. After experimenting with 30 to 40 of the selected features, a decrease in the performance of the system was observed, as the accuracy percentage decreased and the times for training and testing data increased.

3.2.2. Crossover

The second operator of the GA is crossover, which was applied with a probability of 90%. In the crossover, we used single-point crossover, since it has frequently been used by other researchers. The single point was selected randomly such that it was within the first half of the chromosome length. Then the content of the chromosome to the left of the crosspoint was copied, as was the similar position in the offspring, while the content to the right of the crosspoint was switched between parent 1 and parent 2 to produce offspring 1 and offspring 2, respectively, as shown in Figure 5.

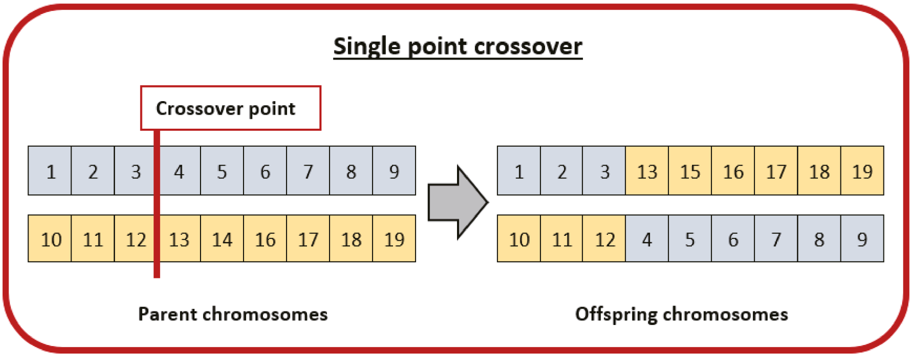


Figure 5. Crossover operation and its effects in generation of offspring.

3.2.3. Mutation

To avoid local convergence, mutation is performed with a very small probability, which was 1% in the present work. In mutation, one gene is selected randomly from the newly generated offspring and replaced with one from the search space, which in the present study was one of the 77 features. During replacement, a check is made to make sure that the newly added gene is not duplicated in the offspring. This is to avoid redundancy of features. An example of the mutation process is illustrated in Figure 6.

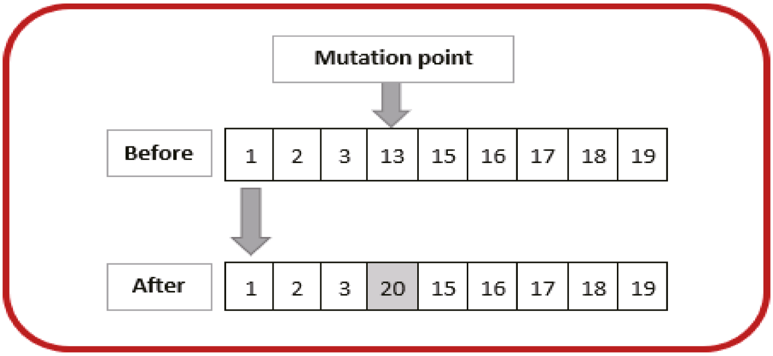


Figure 6. Mutation operation and its effects in generation of offspring.

3.2.4. Improved Fitness Function

The accuracy of classifying the test dataset was used to assess chromosome quality. The fitness function was represented in terms of accuracy. Accuracy is the percentage of the sum of true positive (*TP*) and true negative (*TN*) results divided by the sum of true positive (*TP*), true negative (*TN*), false negative (*FN*), and false positive (*FP*) results. It has been adopted by several researchers as a fitness function (as mentioned in the previous section) to evaluate GA performance. Accuracy was calculated per the following formula:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}, \tag{1}$$

Accuracy cannot be considered a sole metric with which to evaluate a system, since it is not an accurate measure if the dataset is not balanced (both negative and positive classes have a different number of data instances). Hence, the proposed fitness function to be considered in this work was developed using several measures. These measures were F1-score, accuracy, and false positive rate (FPR). F1-score is a metric that considers both precision and recall and is defined as follows:

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2}$$

where precision is the fraction of true positive results among all retrieved results and calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

On the other hand, recall, or true positive rate (TPR), is the fraction of true positive results among all true positive samples in the dataset and calculated as follows:

$$Recall(true\ positive\ rate) = \frac{TP}{TP + FN} \tag{4}$$

The proposed fitness function combined these three metrics in the following formula with different weights for each metric:

$$fitness = \alpha\ F1_{score} + \beta \times \text{accuracy}\ +\ \gamma\ \times \text{TPR} \tag{5}$$

where α, β, and γ are weighting factors that sum to 1. Since F1-score provides high indication to the results, it was assigned the highest value, 0.6, while both accuracy and TPR were considered as secondary measures and assigned 0.2 for each. The optimal value of the fitness function can be obtained through examining all possible combinations of weighting factors, which sum up to 66 possible combinations.

3.2.5. Stopping Criteria

The process of the GA is repetitive. For each iteration, there is a generation, which consists of a set of chromosomes that represent a solution. The maximum number of iterations was set to 100. Therefore, the stopping criteria were either the GA running for 100 iterations or a lack of improvement in the solution obtained. Improvement was measured in terms of the fitness function defined above, so the stopping criterion was a lack of improvement in the obtained accuracy. To define the improvement, we used a threshold. If the difference of accuracy between two consecutive generations was less than the threshold, then the process stopped. The threshold for this system was set to be 0.05.

3.3. Support Vector Machine (SVM)

The support vector machine (SVM) is one of the best machine learning algorithms used for binary classification. IDS can be viewed as binary classification, since transactions are classified as either normal or intrusions regardless of the type of attack.

This work adopted an SVM because of the following advantages. SVMs are highly effective in high-dimensional spaces such as the case we studied here. Even if the number of dimensions is greater than the number of samples, an SVM still produces effective results. SVMs are memory efficient, because in the decision function, they use a subset of training points. SVMs are also versatile, as the decision function can be specified using different kernel functions [27].

3.4. Integration between GA and SVM

The process starts by applying a genetic algorithm to select the needed number of features. At this stage, the preprocessed data (explained later in Section 4.1) is fed into the proposed GA to extract the proper set of features, since the original data consist of a high-dimensional feature set with 77 features, out of which the best features are extracted. In the next stage, the dataset of these features is divided into training and testing sets. After that, a support vector machine is applied to the training set, and then the fitness is calculated. If the fitness is not accepted, the SVM parameters are changed, and SVM is applied again on the training set to calculate the new accuracy. What is meant by not accepted is that the accuracy is less than the threshold. For this work, the threshold was set to the minimum accuracy achieved by other researchers, which was 94.86% [7]. If the results are accepted, they are saved for future comparison with results obtained from other iterations of different feature selections. The next step is to check the total number of iterations; if it is greater than the threshold, then these results, along with the parameters that led to these results, are stored, and the process stops. If the iterations are less than the threshold, the process loops again to the first step. Figure 7 shows the flowchart of the proposed hybrid IDS. In the present work, the GA at the first stage was run for 100 generations. At each generation, a collection of features was examined using the fitness function on that sample. The proposed Hybrid IDS algorithm is presented in Algorithm 2:

Algorithm 2: Hybrid IDS using GA and SVM.

```
Apply Algorithm 1 to select features.
loop
    Create initial generation.
    for i=1 to n /*n is the number of folds*/
        Select 1/n of the population as a test set and (n−1)/n for training.
        Apply SVM to the current training set with specific hyperparameters.
        Evaluate the performance of the results using the fitness function.
    end for
```

Algorithm 2: *Cont.*

```
9:     performance of the current generation= average score of the results of n folds
10:      Save the results of step 6.
11: until the stop criteria are met.
12: Display the best result, along with the hyperparameters for the SVM that produced it.
```

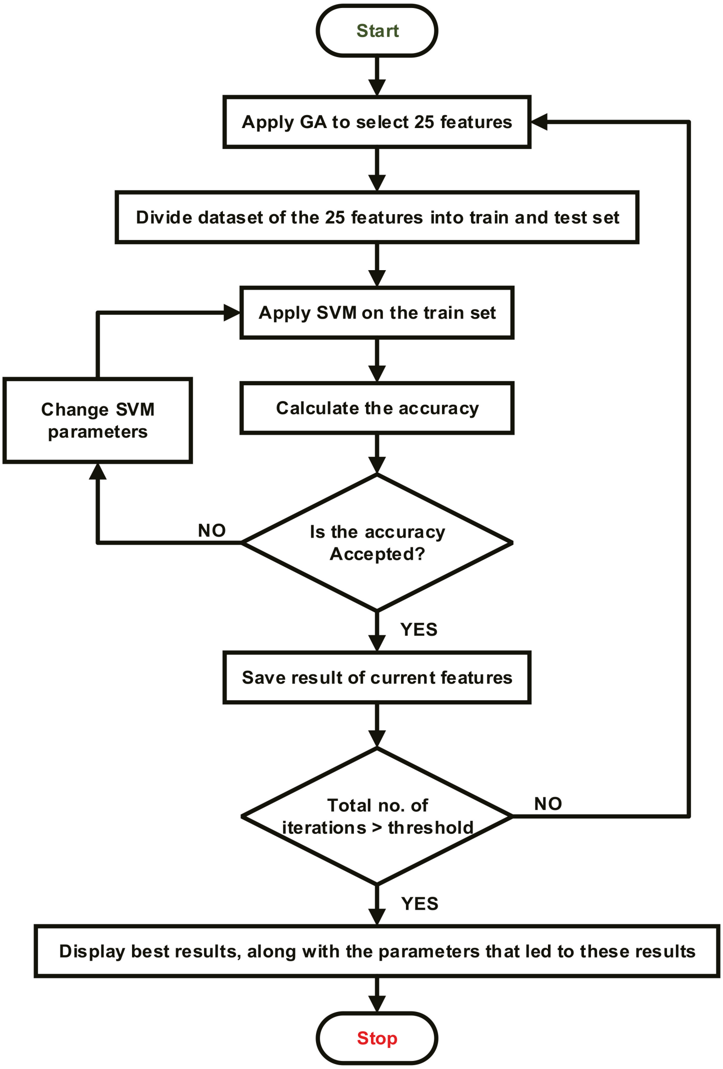

Start
Apply GA to select 25 features
Divide dataset of the 25 features into train and test set
Apply SVM on the train set
Change SVM parameters
Calculate the accuracy
NO
Is the accuracy Accepted?
YES
Save result of current features
Total no. of iterations > threshold
NO
YES
Display best results, along with the parameters that led to these results
Stop

Figure 7. Flowchart of the proposed hybrid IDS.

4. Implementation and Results

This section explains the implementation process. It starts with data preparation, then explains the measures used to evaluate the system performance. Finally, it discusses the obtained results and how they were evaluated using different metrics so they could be compared with existing techniques.

4.1. Experiments Procedure and Results

The conducted experiments had two folds. The first was to find the optimal set of features, while the second was to identify the set of parameters values for the SVM that produce the maximum performance in terms of the proposed fitness function. For the first fold, the GA started with a chromosome length equal to 10, which represents the number of features. For the second fold, the SVA was applied to these 10 features with different parameter values, which are listed in Table 4. The SVA has three basic parameters: kernel, gamma, and degree. The kernel functions used in this work were linear, polynomial, RBF and sigmoid. For linear, the degree is useless, but degree affects the results when the kernel function is polynomial. The final parameter of the SVM that was applied in the present experiment is gamma. It can be either Scale or Auto. The experiment was conducted using 12 different combinations of these parameters, as illustrated in Table 5. For each combination, the GA was executed with a different set of features for a maximum predefined number of generations. The best result of the last generation was considered in this study. The results of this experiment are shown in Table 5. The same type of experiment was conducted for chromosome lengths of 15, 20, 25, 30, 35, and 40, producing the results listed in Tables 6–11. Beyond 40 features, the system experienced unacceptable slowness; hence, these results were not considered in the analysis.

Table 4. List of SVM parameters.

Kernel	Gamma	Degree
Linear	Scale AUTO	3
Poly	Scale AUTO	1, 2, 3
RBF	Scale AUTO	3
Sigmoid	Scale AUTO	3

Table 5. Results of 12 experiments on the proposed hybrid IDS using 10 features and different SVM parameters.

SVM Parameters	Fitness Value
svclassifier = SVC (kemel = ‘linear’, gamma = ‘scale’, degree = 3)	0.994302
svclassifier = SVC (kernel = ‘linear’, gamma = ‘auto’, degree = 3)	0.994302
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = l)	0.994302
svclassifier = SVC (kemel = ‘poly’, gamma = ‘scale’, degree = 2)	0.994542
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 3)	0.994542
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = l)	0.994302
svclassifier = SVC (kemel = ‘poly’, gamma = ‘auto’, degree = 2)	0.994583
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 3)	0.994542
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘scale’, degree = 3)	0.994624
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘auto’, degree = 3)	0.994624
svclassifier = SVC (kemel = ‘sigmoid’, gamma = ‘scale’, degree = 3)	0.991761
svclassifier = SVC (kemel = ‘sigmoid’, gamma = ‘auto’, degree = 3)	0.991947
Average	0.994031

Names of the selected features: Total Backward Packets, Bwd Packet Length Std, Fwd IAT Total, Bwd IAT Mean, Fwd Header Length, URG Flag Count, CWE Flag, Count, Fwd Avg Packets Bulk, Subflow Fwd Bytes, Active Max.

Table 6. Results of 12 experiments on the proposed hybrid IDS using 15 features and different SVM parameters.

15 Features	
SVM Parameters	**Fitness Value**
svclassifier = SVC (kernel = ‘linear’, gamma = ‘scale’, degree = 3)	0.994562
svclassifier = SVQ (kernel = ‘linear’, gamma = ‘auto’, degree = 3)	0.994562
svdassifler = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = l)	0.994281
svdasslfter = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 2)	0.994583
svclassifier = SVC (kernel = ’poly’, gamma = ‘scale’, degree = 3)	0.994583
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = l)	0.994302
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 2)	0.994583
svdassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 3)	0.994583
svdassifier = SVC (kernel = ’rbf’, gamma = ‘scale’, degree = 3)	0.994604
svdassifier = SVC (kernek = ‘rbf’, gamma = ‘auto’, degree = 3)	0.994604
svdassifier = SVC (kernel = ‘sigmoid’, gamma = ‘scale’, degree = 3)	0.992954
svdassifier = SVC (kernel = ‘sigmoid’, gamma = ‘auto’, degree = 3)	0.992954
Average	0.994263

Names of the selected features: Fwd Packet Length Std, Bwd IAT Max, Bwd I AT Min, Fwd PSH Flags, Fwd Header Length, Fwd Packets_s, Bwd Packets_s, ECE Flag Count, Avg Fwd Segment Size, Avg Bwd Segment Size, Fwd Avg Packets_Bulk, Bwd Avg Packets_Bulk, Subflow Fwd Bytes, Subflow Bwd Bytes, Min Seg_Size_Forward.

Table 7. Results of 12 experiments of the proposed Hybrid IDS using 20 features and different SVM parameters.

20 Features		**Confusion Matrix**
SVM Parameters	**Fitness Value**	
svclassifier = SVC (kernel = ‘linear’, gamma = ‘scale’, degree = 3)	1	[[18758 551] [411 79084]]
svdassifier = SVC (kernel = ‘linear’, gamma = ‘auto’, degree = 3)	1	[[18758 551] [411 79084]]
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = l)	1	[[18762 547] [850 78645]]
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 2)	1	[[19215 94] [106 79389]]
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 3)	0.997437	[[33494 193] [47 340]]
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = l)	1	[[33429 258] [48 339]]
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 2)	1	[[33487 200] [47 340]]
svclassifier = SVC (kernel-‘poly’, gamma = ‘auto’, degree = 3)	0.999979	[[33686 1] [368 19]]
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘scale’, degree = 3)	0.994901	[[19228 81] [61 79434]]
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘auto’, degree = 3)	0.998482	[[33521 166] [49 338]]

Table 7. *Cont.*

20 Features		**Confusion Matrix**
SVM Parameters	**Fitness Value**	
svclassifier = SVC (kernel = ‘sigmoid’, gamma = ‘scale’, degree = 3)	0.998649	[[33399 288]
svclassifier = SVC (‘kernel-sigmoid’, gamma = ‘auto’, degree = 3)	0.99344	[233 154]]
Average	0.998574	[[33436 251]

Names of the selected features: Total Length of Bwd Packets, Bwd Packet Length Mean, Flow IAT Mean, Flow IAT Max, Fwd IAT Std, Bwd IAT Mean, Bwd IAT Max, Bwd Header Length, Fwd Packets_s, PSH Flag Count, URG Flag Count, ECE Flag Count, Avg Bwd Segment Size, Fwd Avg Bytes Bulk, Bwd Avg Packets Bulk, Subflow Bwd Bytes, Init Win Bytes Backward, Active Min, Idle Mean, Idle Max.

Table 8. Results of 12 experiments on the proposed hybrid IDS using 25 features and different SVM parameters.

25 Features	
SVM Parameters	**Fitness Value**
svclassifier = SVC (kernel = ‘linear’, gamma = ‘scale’, degree = 3)	0.994427
svclassifier = SVC (kemel = ‘linear’, gamma = ‘auto’, degree = 3)	0.994427
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = l)	0.99426
svclassifier = SVC (kemel = ‘poly’, gamma = ‘scale’, degree = 2)	0.995953
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 3)	0.996404
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = l)	0.99426
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 2)	0.995953
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 3)	0.996341
svclassifier = SVC (kemel = ‘rbf’, gamma = ‘scale’, degree = 3)	0.996762
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘auto’, degree = 3)	0.996762
svclassifier = SVC (kernel = ‘sigmoid’, gamma = ‘scale’, degree = 3)	0.986537
svclassifier = SVC (kemel = ‘sigmoid’, gamma = ‘auto’, degree = 3)	0.986537
Average	0.994052

Names of the selected features: Flow Duration, Total Fwd Packets, Fwd Packet Length Std, Flow IAT Min, Fwd IAT Max, Fwd IAT Min, Bwd IAT Total, Fwd PSH, Flags, Max Packet Length, Packet Length Variance, SYN Flag Count, RST Flag Count, URG Flag Count, CWE Flag Count, ECE Flag, Count, Fwd Header Length, Fwd Avg Bulk Rate, Subflow Fwd Bytes, Init_Win_Bytes_Forward, Init_Win_Bytes_Backward, Act Data Pkt Fwd, Active Min, Idle Mean, Idle Max, Idle Min.

Table 9. Results of 12 experiments on the proposed hybrid IDS using 30 features and different SVM parameters.

30 Features	
SVM Parameters	**Fitness Value**
svclassifier = SVC (kemel = ‘linear’, gamma = ‘scale’, degree = 3)	0.994583
svclassifier = SVC (kernel = ‘linear’, gamma = ‘auto’, degree = 3)	0.994583
svclassifier = SVC (kemel = ‘poly’, gamma = ‘scale’, degree = l)	0.994583
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 2)	0.994624
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 3)	0.994539
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = l)	0.994583
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 2)	0.994624
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 3)	0.996804
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘scale’, degree = 3)	0.99458

Table 9. *Cont.*

30 Features	
SVM Parameters	**Fitness Value**
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘auto’, degree = 3)	0.99458
svclassifier = SVC (kernel = ‘sigmoid’, gamma = ‘scale’, degree = 3)	0.987238
svclassifier = SVC (kernel = ‘sigmoid’, gamma = ‘auto’, degree = 3)	0.987363
Average	0.993557

Names of the selected features: Total Fwd Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Bwd Packet Length Max, Bwd Packet Length Min, Flow IAT Mean, Fwd IAT Total, Fwd IAT Std, Fwd IAT Max, Bwd IAT Mean, Bwd IAT Std, Fwd PSH Flags, Bwd PSH Flags, Bwd Header Length, Fwd Packets_s, Packet Length Variance, SYN Flag Count, PSH Flag Count, CWE Flag Count, ECE Flag Count, DownJJp Ratio, Average Packet Size, Avg Bwd Segment Size, Fwd Avg Packets_Bulk, Bwd Avg Bytes_Bulk, Bwd Avg Bulk Rate, Subflow Bwd Bytes, Active Max, Active Min, Idle Mean.

Table 10. Results of 12 experiments on the proposed hybrid IDS using 35 features and different SVM parameters.

35 Features	
SVM Parameters	**Fitness Value**
svclassifier = SVC (kernel = ‘linear’, gamma = ‘scale’, degree = 3)	0.992532
svclassifier = SVC (kemel = ‘linear’, gamma = ‘auto’, degree = 3)	0.992532
svclassifier = SVC (kemeI = ‘poly’, gamma = ‘scale’, degree = l)	0.996033
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 2)	0.996951
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 3)	0.996951
svclassifier = SVC (kemel = ‘poly’, gamma = ‘auto’, degree = l)	0.995778
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 2)	0.992633
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 3)	0.992574
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘scale’, degree = 3)	0.993822
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘auto’, degree = 3)	0.993808
svclassifier = SVC (kernel = ‘sigmoid’, gamma = ‘scale’, degree = 3)	0.986881
svclassifier = SVC (kernel = ‘sigmoid’, gamma = ‘auto’, degree = 3)	0.992652
Average	0.993808

Names of the selected features: Total Length of Fwd Packets, Fwd Packet Length Max, Fwd Packet Length Min, Fwd Packet Length Mean, Bwd Packet Length Min, Bwd Packet Length Std, Flow IAT Std, Flow IAT Max, Flow IAT Min, Fwd IAT Std, Bwd IAT Std, Fwd URG Flags, Fwd Header Length, Bwd Header Length, Fwd Packets_s, Bwd Packets_s, Min Packet Length, Packet Length Variance, SYN Flag Count, Average Packet Size, Avg Bwd Segment Size, Fwd Avg Bytes_Bulk, Fwd Avg Packets_Bulk, Fwd Avg Bulk Rate, Bwd Avg Bulk Rate, Subflow Fwd Packets, Subflow Bwd Bytes, Init_Win_Bytes_Forward, Init_Win_Bytes_Backward, Min_Seg_Size_Forward, Active Mean, Active Std, Active Min, Idle Mean, Idle Std.

These experiments produced the following observations: from 10 to 15 features the results were very close, and the training and testing times of the data were adequate. From 25 to 40 features, a decrease in the performance of the system was observed, as the accuracy percentage decreased, and the time of training and testing increased.

To benchmark the performance of the proposed system with other systems using the KDD CUP 99 dataset, another experiment was conducted using the same 12 combinations of SVM parameters. The results of this experiment are shown in Table 12. Analysis of these experiments is presented in the next subsection.

Table 11. Results of 12 experiments on the proposed hybrid IDS using 40 features and different SVM parameters.

40 Features	
SVM Parameters	**Fitness Value**
svclassifier = SVC (kernel = ‘linear’, gamma = ‘scale’, degree = 3)	0.995964
svclassifier = SVC (kemel = ‘linear’, gamma = ‘auto’, degree = 3)	0.995964
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = l)	0.989691
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 2)	0.991477
svclassifier = SVC (kemel = ‘poly’, gamma = ‘scale’, degree = 3)	0.995303
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = l)	0.99463
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 2)	0.995839
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 3)	0.999979
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘scale’, degree = 3)	0.996609
svclassifier = SVC (kernel = ‘rbf’, gamma = ‘auto’, degree = 3)	0.996547
svclassifier = SVC (kemel = ‘sigmoid’, gamma = ‘scale’, degree = 3)	0.991734
svclassifier = SVC (kernel = ‘sigmoid’, gamma = ‘auto’, degree = 3)	0.994776
Average	0.994876

Names of the selected features: Flow Duration, Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length Max, Bwd Packet Length Max, Bwd Packet Length Std, Flow IAT Max, Flow lATMin, Fwd IAT Max, Bwd IATTotal, Bwd lATStd, Bwd lATMin, Fwd PSH Flags, Bwd PSH Flags, Bwd URG Flags, Fwd Header Length, Bwd Header Length, Fwd Packets_s, Bwd Packets_s, Packet Length Mean, Packet Length Variance, SYN Flag Count, RST Flag Count, PSH Flag Count, URG Flag Count, CWE Flag Count, ECE Flag Count, Average Packet Size, Fwd Avg Packets_Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes_Bulk, Bwd Avg Bulk Rate, Subflow Fwd Packets, Act Data Pkt Fwd, Min Seg Size Forward, Active Std, Active Min, Idle Mean, Idle Max, Idle Min.

Table 12. Results of 12 experiments of the proposed Hybrid IDS using 15 features from (KDD CUP99) dataset.

15 Features	
SVM Parameters	**Fitness Value**
svclassifier = SVC (kernel= ‘linear’, gamma = ‘scale’, degree = 3)	0.980213
svclassifier = SVC (kernel = ‘linear’, gamma = ‘auto’, degree = 3)	0.980213
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = l)	0.972088
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 2)	0.985217
svclassifier = SVC (kernel = ‘poly’, gamma = ‘scale’, degree = 3)	0.992792
svclassifier = SVC (kemel = ‘poly’, gamma = ‘auto’, degree = l)	0.972032
svclassifier = SVC (kernel = ‘poly’, gamma = ‘auto’, degree = 2)	0.985074
svclassifier = SVC (kemel = ‘poly’, gamma = ‘auto’, degree = 3)	0.991780
svclassifier = SVC (kernel = ‘rbf’, gamma = scale’, degree = 3)	0.992975
svclassifier = SVC (kemel = ‘rbf’, gamma = ‘auto’, degree = 3)	0.992650
svclassifier = SVC (kernel = ‘sigmoid’, gamma = ‘scale’, degree = 3)	0.893446
svclassifier = SVC (kernel = ‘sigmoid’, gamma = ‘auto’, degree = 3)	0.886727
Average	0.968767

Name of the selected features: service, flag, src_bytes, num_failed_logins, logged_in, lnum_file_creations, lnum_outbound_cmds, count, srv_count, rerror_rate, same srv rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst host rerror rate, dst host srv rerror rate.

4.2. Analysis of Results

Different types of SVM kernel functions were applied. It was observed that if the kernel was linear, then the accuracy did not depend on any other parameters (neither gamma nor degree). If the kernel was polynomial and gamma was scale, then the accuracy depended on the degree. When the degree was two or three, the SVM produced higher accuracy than if the degree was one in most cases. In terms of the number of features, it was noted that the highest ever scored fitness, which was 100%, was obtained when the number of features was 20. This high performance was achieved when SVM kernel function was linear, or when the kernel function was polynomial and the degree was either one or two. In addition, the highest average of all 12 combinations was achieved when the number of features was 20 as well. Graphically, this is illustrated in Figure 8. This showed that the optimal number of features was 20, which was 25.6% of features in the dataset. These features are listed in Table 7. Achieving the optimal performance of 100% using an only quarter of features proved that the proposed method could perform well using a limited number of features. On the other hand, one could conclude that the dataset had many low-value features that could be eliminated from the dataset safely. However, this conclusion needs to be further examined using other ML techniques.

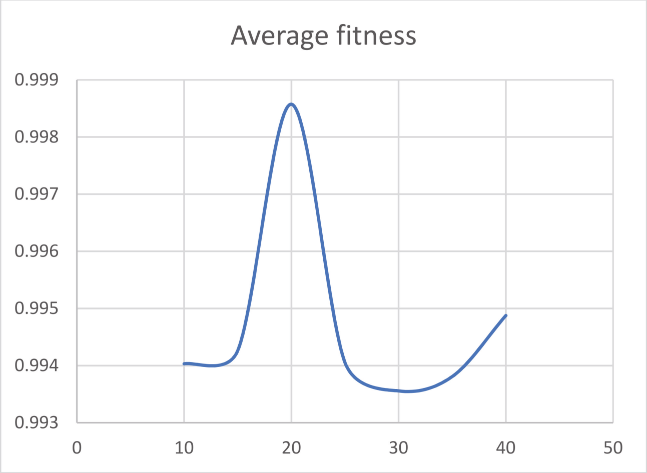


Figure 8. Average fitness per number of features.

Comparing the obtained results using the CICIDS2017 dataset with those of the other hybrid IDS mentioned in Section 2, which used the KDD CUP 99 dataset, it was noted that [4] achieved a 96.38% detection rate while our system achieved a maximum of 100%, which means that the proposed system achieved an improvement of 3.32%. Reference [5] achieved a 95.26% detection rate compared with the 99.50% achieved in the present study, which means that the proposed system achieved an improvement of 4.74%. Reference [6] achieved a 95.17% detection rate, which was lower than that obtained in the present study by 4.83%. Reference [7] achieved a 94.86% detection rate, which was 5.14% less than that obtained by the proposed hybrid IDS. These results are summarized in Table 13.

Table 13. Summary of results analysis using the CICIDS2017 dataset.

Authors	Detection Rate (DR)	Proposed Method (DR)	Achieved Improvement (%)
M. Moukhafi et al., 2018 [4]	96.38%	100%	3.32
F. Kuang et al., 2018 [5]	95.26%		4.74
Al-yaseen et al., 2018 [6]	95.17%		4.83
W. Feng et al., 2014 [7]	94.86%		5.14

The proposed system was also tested with the KDD CUP 99 dataset to compare the present results with previous works that used the same dataset in their works. It was noted that [4] achieved a 96.38% detection rate, while our system achieved a maximum detection rate of 99.3% using the same dataset, which means the proposed system achieved an improvement of 3.03%. Reference [5] achieved a 95.26% detection rate, while the proposed hybrid IDS system achieved a maximum detection rate of 99.65%, which means that the proposed system achieved an improvement of 4.24%. Reference [6] achieved a 95.17% detection rate, which is lower than that obtained by the proposed system by

4.33%. Referenced [7] achieved a 94.86% of detection rate, which is 4.68% lower than that obtained by the proposed hybrid IDS. Compared with other IDS systems that applied fuzzy vectorized GA as a core technique, it was noted that [39] used a vectorized fitness function on the NSL-KDD dataset with 42 features. The NSL-KDD dataset is the enhanced version of KDD CUP 99. In the experiment in [39], six different models were applied, four of which achieved accuracy ranging between 91.86 and 95.53%. On the other hand, two models achieved accuracies of 99.18% and 99.02%. The last two scores were much closer to the achieved results in this work; they were less by only 0.12% and 0.28%, respectively. Another work that applied GA is [47]. The authors developed a GPSVM, which combined both genetic programming and an SVM. The detection rate varied depending on the type of attack. The maximum detection rate achieved was 97.59%, for detecting U2R attacks. It was noted that this detection rate was 1.75% lower than that obtained by our system. Considering the results obtained by a recent study, [48], in which intrusion was detected using a real-time sequential deep extreme learning machine, said machine achieved a maximum accuracy rate of 93.58% when applied on a fused dataset (NSL-KDD and KDD CUP 99). One of the recent studies is [49]. The system therein used a deep extreme learning machine (DELM) to identify any malice or intrusion. Its accuracy using KDD CUP 99 was 94.6%, which was better than its accuracy using NSL-KDD by 0.69%. It is worth pointing that our proposed system outperformed the DELM by up to 5.47%. Table 14 summarizes these results.

Table 14. Summary of results analysis using the KDD CUP 99 and NSL-KDD datasets.

Authors	Accuracy Rate (%)	Dataset	Achieved Improvement (%)
Proposed Hybrid IDS (this work)	99.3	KDD CUP 99	—
M. Moukhafi et al., 2018 [4]	96.38		3.03
F. Kuang et al., 2018 [5]	95.26		4.24
Al-yaseen et al., 2018 [6]	95.17		4.34
W. Feng et al., 2014 [7]	94.86		4.68
Khan et al., 2021 [49]	94.6		4.97
Bhattacharjee et al., 2017 (f1) [39]	99.18	NSL-KDD	0.12
Bhattacharjee et al., 2017 (f2) [39]	99.02		0.28
Pozi et al., 2016 [47]	97.59		1.75
Khan et al., 2021 [49]	93.91		5.74
Khan et al., 2021 [48]	93.58	KDD CUP 99 and NSL-KDD	6.11

5. Conclusions

In this work, we propose a novel hybrid intrusion detection system to secure data in cloud computing based on an improved genetic algorithm (GA) and support vector machine (SVM) algorithm to create an intelligent IDS model. The improved GA is characterized by a newly developed fitness function that is used to evaluate the performance of the hybrid IDS. The fitness function combines three measures: F1-score, accuracy, and TPR, with different weights for each measure. This system was examined using two datasets: the newly developed dataset CICIDS2017, which consists of normal and most up-to-date common attacks, and the well-known dataset KDD CUP 99. The brilliance of the proposed system is that the GA and SVM are executed simultaneously to achieve two objectives at once, which are obtaining the best subset of 79 features with maximum accuracy. The SVM is used to classify data into benign and abnormal using different values for its hyperparameters, including the kernel function, gamma, and degree. Folding of the dataset is applied to change the training and testing dataset to examine the behavior of the system on these kinds of data. The results were analyzed and compared with similar works that applied GA and SVM for IDS. The results showed that the proposed model outperformed these works in terms of accuracy by a maximum of 5.14% and a minimum of 3.32% using CICIDS2017, a maximum of 4.97% and a minimum of 3.03% using the KDD CUP 99 dataset, and a maximum of 5.74% and a minimum of 0.12% using the NSL-KDD dataset. This system proved its efficiency through the results obtained. To generalize the results, further experiments must be conducted. These must include using even larger datasets, using different datasets, trying different machine learning algorithms or combinations of ML algorithms, and trying more hyperparameters and parameter values for these ML algorithms.

Author Contributions: A.A. Conceptualization, Methodology, software, project administration, supervision, writing—review and editing. F.A. Formal analysis, investigation, Methodology, data curation, software, writing—original draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Medical Data in the Crosshairs: Why Is Healthcare an Ideal Target? 14 August 2021. Available online: https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/medical-data-in-the-crosshairs-why-is-healthcare-an-ideal-target (accessed on 15 May 2020).
2. Conaty-Buck, S. Cybersecurity and healthcare records. *Am. Nurse Today* **2017**, *12*, 62–64.
3. eGOVERNMENT, Cloud Computing Initiatives. 22 April 2021. Available online: https://www.bahrain.bh/ (accessed on 15 July 2021).
4. Moukhafi, M.; El Yassini, K.; Bri, S. A novel hybrid GA and SVM with PSO feature selection for intrusion detection system. *Int. J. Adv. Sci. Res. Eng.* **2018**, *4*, 129–134. [CrossRef]
5. Kuang, F.; Xu, W.; Zhang, S. A novel hybrid KPCA and SVM with GA model for intrusion detection. *Appl. Soft Comput. J.* **2014**, *18*, 178–184. [CrossRef]
6. Al-Yaseen, W.L.; Othman, Z.A.; Nazri, M.Z.A. Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Syst. Appl.* **2017**, *67*, 296–303. [CrossRef]
7. Feng, W.; Zhang, Q.; Hu, G.; Huang, J.X. Mining network data for intrusion detection through combining SVMs with ant colony networks. *Future Gener. Comput. Syst.* **2014**, *37*, 127–140. [CrossRef]
8. Ambusaidi, M.A.; He, X.; Nanda, P.; Tan, Z. Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE Trans. Comput.* **2016**, *65*, 2986–2998. [CrossRef]
9. Mustapha, B.; Salah, E.H.; Mohamed, I. A two-stage classifier approach using RepTree algorithm for network intrusion detection. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **2017**, *8*, 389–394.
10. Tuan, A.; McLernon, D.; Mhamdi, L.; Zaidi, S.A.R.; Ghogho, M. Intrusion Detection in SDN-Based Networks: Deep Recurrent Neural Network Approach. In *Advanced Sciences and Technologies for Security Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 175–195.
11. Nguyen, K.K.; Hoang, D.T.; Niyato, D.; Wang, P.; Nguyen, D.; Dutkiewicz, E. Cyberattack detection in mobile cloud computing: A deep learning approach. In Proceedings of the IEEE Wireless Communications and Networking Conference, Barcelona, Spain, 15–18 April 2018.
12. He, D.; Qiao, Q.; Gao, Y.; Zheng, J.; Chan, S.; Li, J.; Guizani, N. Intrusion detection based on stacked autoencoder for connected healthcare systems. *IEEE Netw.* **2019**, *33*, 64–69. [CrossRef]
13. Mudzingwa, D.; Agrawal, R. A study of methodologies used in intrusion detection and prevention systems (IDPS). In Proceedings of the IEEE Southeastcon, Orlando, FL, USA, 15–18 March 2012.
14. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
15. Ludinard, R.; Totel, É.; Tronel, F.; Nicomette, V.; Kaâniche, M.; Alata, É.; Bachy, Y. Detecting attacks against data in web applications. In Proceedings of the 7th International Conference on Risks and Security of Internet and Systems (CRiSIS), Cork, Ireland, 10–12 October 2012.
16. Li, X.; Xue, Y.; Malin, B. Detecting anomalous user behaviors in workflow-driven web applications. In Proceedings of the IEEE Symposium on Reliable Distributed Systems, Irvine, CA, USA, 8–11 October 2012; pp. 1–10.
17. Le, M.; Stavrou, A.; Kang, B.B. DoubleGuard: Detecting intrusions in multitier web applications. *IEEE Trans. Dependable Secur. Comput.* **2012**, *9*, 512–525. [CrossRef]

18. Nascimento, G.; Correia, M. Anomaly-based intrusion detection in software as a service. In Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), Hong Kong, China, 27–30 June 2011; pp. 19–24.
19. Ariu, D. Host and Network Based Anomaly Detectors for HTTP Attacks. Ph.D. Thesis, University of Cagliari, Cagliari, Italy, 2010.
20. Gimenez, C.; Villaegas, A.; Alvarez, G. An Anomaly-Based Approach for Intrusion Detection in Web Traffic. *J. Inf. Assur. Secur.* **2010**, *5*, 446–454.
21. Saraniya, G. Securing the Network Using Signature Based IDS in Network IDS. *Shodhshauryam Int. Sci. Refereed Res. J.* **2019**, *2*, 99–101.
22. Gao, W.; Morris, T. On Cyber Attacks and Signature Based Intrusion Detection for Modbus Based Industrial Control Systems. *J. Digit. Forensics Secur. Law* **2014**, *9*, 37–56. [CrossRef]
23. Uddin, M.; Rehman, A.A.; Uddin, N.; Memon, J.; Alsaqour, R.; Kazi, S. Signature-based multi-layer distributed intrusion detection system using mobile agents. *Int. J. Netw. Secur.* **2013**, *15*, 97–105.
24. Kumar, U.; Gohil, B.N. A Survey on Intrusion Detection Systems for Cloud Computing Environment. *Int. J. Comput. Appl.* **2015**, *109*, 6–15. [CrossRef]
25. Lewis, P.M. The characteristic selection problem in recognition system. *IEEE Trans. Inf. Theory* **1962**, *8*, 171–178. [CrossRef]
26. Shakya, V.; Makwana, R.R.S. Feature selection-based intrusion detection system using the combination of DBSCAN, K-Mean++ and SMO algorithms. In Proceedings of the 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, India, 11–12 May 2017; pp. 928–932. [CrossRef]
27. Almomani, O. A Feature Selection Model for Network Intrusion Detection System Based on PSO, GWO, FFA and GA Algorithms. *Symmetry* **2020**, *12*, 1046. [CrossRef]
28. Gul, A.; Adali, E. A feature selection algorithm for IDS. In Proceedings of the 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 5–8 October 2017; pp. 816–820. [CrossRef]
29. Gharaee, H.; Hosseinvand, H. A new feature selection IDS based on genetic algorithm and SVM. In Proceedings of the 2016 8th International Symposium on Telecommunications (IST), Tehran, Iran, 27–28 September 2016; pp. 139–144. [CrossRef]
30. Parimala, G.; Kayalvizhi, R. An Effective Intrusion Detection System for Securing IoT Using Feature Selection and Deep Learning. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 27–29 January 2021; pp. 1–4. [CrossRef]
31. Hakim, L.; Fatma, R. Influence Analysis of Feature Selection to Network Intrusion Detection System Performance Using NSL-KDD Dataset. In Proceedings of the 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), Jember, Indonesia, 16–17 October 2019; pp. 217–220. [CrossRef]
32. Ahmadi, S.S.; Rashad, S.; Elgazzar, H. Efficient Feature Selection for Intrusion Detection Systems. In Proceedings of the 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 10–12 October 2019; pp. 1029–1034. [CrossRef]
33. Wang, W.; Du, X.; Wang, N. Building a Cloud IDS Using an Efficient Feature Selection Method and SVM. *IEEE Access* **2019**, *7*, 1345–1354. [CrossRef]
34. Tao, P.; Sun, Z.; Sun, Z. An Improved Intrusion Detection Algorithm Based on GA and SVM. *IEEE Access* **2018**, *6*, 13624–13631. [CrossRef]
35. Hsu, C.-W.; Chang, C.-C.; Lin, C.-J. *A Practical Guide to Support Vector Classification*; Department of Computer Science, National Taiwan University: Taipei, Taiwan, 2003.
36. Rai, N.; Rai, K. Genetic algorithm-based intrusion detection system. *Int. J. Comput. Sci. Inf. Technol.* **2014**, *5*, 4952–4957.
37. Ojugo, A.A.; Eboka, A.O.; Okonta, O.E.; Yoro, R.E.; Aghware, F.O. Genetic Algorithm Rule-Based Intrusion Detection System (GAIDS). *J. Emerg. Trends Comput. Inf. Syst.* **2012**, *3*, 1182–1194.
38. Bhattacharjee, P.S. Intrusion Detection System for NSL-KDD Data Set using Vectorised Fitness Function in Genetic Algorithm. *Adv. Comput. Sci. Technol.* **2017**, *10*, 235–246.
39. Kalavadekar, M.P.N.; Sane, S.S. Building an Effective Intrusion Detection System using Genetic Algorithm based Feature Selection. *Int. J. Comput. Sci. Inf. Secur.* **2018**, *16*, 97–110.
40. Gong, R.H.; Zulkernine, M.; Abolmaesumi, P. A software implementation of a genetic algorithm-based approach to network intrusion detection. In Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network, Towson, MD, USA, 23–25 May 2005; Volume 2005, pp. 246–253.
41. Pawar, S.N.; Bichkar, R.S. Genetic algorithm with variable length chromosomes for network intrusion detection. *Int. J. Autom. Comput.* **2015**, *12*, 337–342. [CrossRef]
42. Agarwal, B.; Mittal, N. Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques. *Procedia Technol.* **2012**, *6*, 996–1003. [CrossRef]
43. Serpen, G.; Aghaei, E. Host-based misuse intrusion detection using PCA feature extraction and kNN classification algorithms. *Intell. Data Anal.* **2018**, *22*, 1101–1114. [CrossRef]
44. Praveen, K.; Rajendran, M.; Rajesh, R. Healthcare Systems Hybrid Intrusion Detection Algorithm for Private Cloud. *Indian J. Sci. Technol.* **2015**, *48*, 325–329.

45. CICIDS2017, Intrusion Detection Evaluation Dataset, Last Update (2020). Available online: https://www.kaggle.com/cicdataset/cicids2017 (accessed on 25 February 2021).
46. KDD Cup 1999 Data. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 23 February 2021).
47. Pozi, M.; Sulaiman, M.N.; Mustapha, N.; Perumal, T. Improving Anomalous Rare Attack Detection Rate. *Neural Process. Lett.* **2015**, *44*, 279–290. [CrossRef]
48. Khan, M.A.; Ghazal, T.M.; Lee, S.W.; Rehman, A. Data Fusion-Based Machine Learning Architecture for Intrusion Detection. *Comput. Mater. Contin.* **2021**, *70*, 3399–3413. [CrossRef]
49. Khan, A.; Abbas, S.; Rehman, A.; Saeed, Y.; Zeb, A.; Uddin, M.I.; Nasser, N.; Ali, A. A machine learning approach for blockchain-based smart home networks security. *IEEE Netw.* **2021**, *35*, 223–229. [CrossRef]

symmetry MDPI

Article

Quantum Game Application to Recovery Problem in Mobile Database

Magda M. Madbouly [1], Yasser F. Mokhtar [2] and Saad M. Darwish [1,*]

[1] Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Alexandria 21526, Egypt; mmadbouly@hotmail.com
[2] IT Department, Egyptian Petrochemicals Co., Alexandria 4669, Egypt; yasserfakhry@hotmail.com
* Correspondence: saad.darwish@alexu.edu.eg

Abstract: Mobile Computing (MC) is a relatively new concept in the world of distributed computing that is rapidly gaining traction. Due to the dynamic nature of mobility and the limited bandwidth available on wireless networks, this new computing environment for mobile devices presents significant challenges in terms of fault-tolerant system development. As a consequence, traditional fault-tolerance techniques are inherently inapplicable to these systems. External circumstances often expose mobile systems to failures in communication or data storage. In this article, a quantum game theory-based recovery model is proposed in the case of a mobile host's failure. Several of the state-of-the-art recovery protocols are selected and analyzed in order to identify the most important variables influencing the recovery mechanism, such as the number of processes, the time needed to send messages, and the number of messages logged-in time. Quantum game theory is then adapted to select the optimal recovery method for the given environment variables using the proposed utility matrix of three players. Game theory is the study of mathematical models of situations in which intelligent rational decision-makers face conflicting interests (alternative recovery procedures). The purpose of this study is to present an adaptive algorithm based on quantum game theory for selecting the most efficient context-aware computing recovery procedure. The transition from a classical to a quantum domain is accomplished in the proposed model by treating strategies as a Hilbert space rather than a discrete set and then allowing for the existence of linear superpositions between classical strategies; this naturally increases the number of possible strategic choices available to each player from a numerable to a continuous set. Numerical data are provided to demonstrate feasibility.

Keywords: mobile computing; quantum game theory; decision making; mobile database recovery

Citation: Madbouly, M.M.; Mokhtar, Y.F.; Darwish, S.M. Quantum Game Application to Recovery Problem in Mobile Database. *Symmetry* **2021**, *13*, 1984. https://doi.org/10.3390/sym13111984

Academic Editors: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin, Jen-Chun Lee and Sergei D. Odintsov

Received: 27 July 2021
Accepted: 15 October 2021
Published: 20 October 2021

1. Introduction

The mobile database system (MDS) is a client/server database management system provided via the internet that allows for the mobility of the whole processing environment. While the database itself may be static and spread across many sites, data processing nodes such as laptops, PDAs, and cell phones may be mobile and access required data from any place and at any time. A mobile host (MH) operating a client-server application may rapidly fail due to limited network resources. Client-server application failure recovery needs significant attention due to the scope of its usage. When applied to the mobile computing environment, traditional recovery methods such as checkpointing, logging, and rollback recovery suffer from many limitations [1–5].

Using checkpoint and message logging techniques, the mobile application may roll back to the last reliably stored state and resume execution with recovery assurances. Existing approaches operate under the assumption that MH disk storage is insecure and store checkpoint and log data at base stations [6–9]. The process for a mobile checkpoint may be coordinated or uncoordinated. To maintain a consistent and recoverable global checkpoint, distributed applications need MHs to coordinate their local checkpoints. Because the MH can independently checkpoint its local state, uncoordinated protocols are preferred for mobile applications [7–9]. Recovery methods that are not coordinated are either non-logging or logging [4,5]. No-logging MH must generate a new checkpoint whenever the application's state changes. The logging method periodically generates checkpoints and records all write events between them. When an MH attempts to recover from a failure, it makes use of the checkpoint and any previously stored log data. The survey [1,9] compared performance with and without logging.

Numerous factors have an effect on recovery [1,3]: (1) the failure of the MH, for example, due to a bad wireless connection or inadequate battery capacity, is entirely random. If further failures occur, the transaction must roll back whenever MH recovers from a failure, increasing the total execution duration of the transaction; (2) Log Size: data transmission consumes twice the amount of energy required for data receipt. As a result, only important write events should be recorded to maintain a short log; (3) Memory Constraints: the base station controller (BSC) may need a significant amount of memory in order to store the log file for each MH. Calculating the average memory need for logs of different lengths and recovery techniques is essential; (4) Time required for recovery: the time required to recover a process after a failure varies according to the recovery method used and the technology used to capture write events; (5) Cost of log retrieval: the cost of reclaiming log information after the failure of a transaction is related to the amount of log dispersion. When a log is dispersed over several places, the costs of retrieval and recovery increase [10–13].

At present, academics are addressing wireless communication problems using game-theoretic methods [14,15]. In comparison to more conventional approaches, game theory offers a number of benefits. Game theory is concerned with a range of problems involving the strategic interaction of many individuals with conflicting goals in a competition. As a consequence, game theory is an inherently useful tool for describing the rational behavior of many players. Second, game theory may be utilized to model agent-agent interactions, to analyze equilibrium, and to develop distributed algorithms. Additionally, game theory is capable of analyzing hundreds of potential outcomes prior to determining the best course of action.

The Nash Equilibrium (NE) is the state of affairs in which no player can unilaterally enhance their reward, while the Pareto Optimum (PO) is the state of affairs in which no player can unilaterally increase their reward without impairing the advantage of another player. Both are optimal for the individual player, but the latter is usually preferable for the whole team [14,15].Quantum game theory has developed as a paradigm for analyzing the competitive flow of quantum information in the recent past [16]. The phrase "quantum game"refers to novel applications of quantum information processing, such as competitive agent interactions. In contrast to conventional communication, applications may be created based on the interactions of entangled agents. Competitive von Neumann games, such as quantum auctions and voting, are enabled by entangled resources. These are in contrast to cooperative games, which allow agents to interact directly or indirectly via a third party. Entanglement is a quantum resource that may be utilized to optimize known game-theoretic equilibrium outcomes [16–19]. Figure 1 depicts the classical and quantum versions of a four-player game. Although players in the quantum version use an entangled state as a resource, neither version allows for player communication. The bar graph depicts the equilibrium payoffs for the minority game, while the quantum case's payoffs were determined experimentally using our technology.

Motivation and Contribution

In most mobile application recovery methods, environmental factors were not taken into account as potential influencers on the recovery process. As a result, using recovery methods in the real world is challenging [20]. The goal is to create a strategy that optimizes achievement by using the most effective recovery techniques available in light of the present circumstances. We selected quantum game theory over conflict analysis or interactive decision theory because it enables us to compare the recovery techniques' available alternatives. The suggested model makes a significant contribution by enabling effective MDS recovery treatment by using a new smart strategy centered on players (various recovery procedures) inside the quantum game theory paradigm as a decision-maker for choosing the most efficient recovery process. Because the critical problem is not to choose one of the well-known recovery techniques, but to choose the strategy that is most appropriate in light of the changes made by the operating environment, which is often vague and unpredictable. In this respect, the present study will guarantee that the optimal approach to recovery through the quantum game theory model is selected based on its key parameters. This research examines many different types of recovery methods. These methods show the effect of different factors on the protocol's complicated efficiency. The suggested model demonstrates a high degree of adaptability via the use of cutting-edge recovery methods capable of substantially increasing performance.

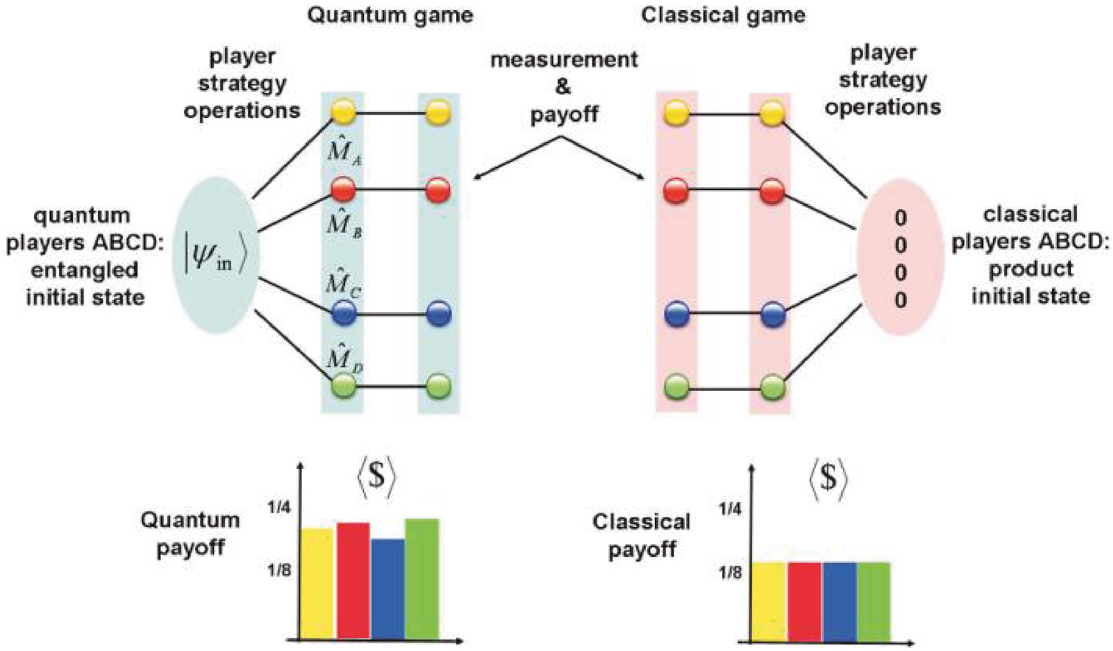


Figure 1. Diagram depicting the information flow in a four-player classical game (**right**) and a four-player quantum game (**left**).

To our knowledge, this is the first time that quantum game theory has been used to simulate recovery in a mobile database system. This work extends our conference paper [21] significantly by using quantum game theory rather than classical game theory to successfully deal with NE in order to increase the payoffs of well-known game-theoretic equilibrium. Additionally, three recovery algorithms were employed as players in the quantum game theory, rather than two in our earlier work, to allow a greater number of algorithms to join the competition, which enhances decision-making effectiveness under varying environmental circumstances. The transition from two-player to multiplayer games unquestionably results in more dynamic problem solving, and this mode is the most accurate representation of reality. Single-game dynamics are generally constrained, and trajectories may show a variety of limiting characteristics. When additional participants are involved, both qualitative and statistical dynamics may change. Analytic techniques are used to assess the suggested model's performance, and the results are given later in the article.

A basic notion in computer science, especially in the theoretical realm, is symmetry. Symmetry is crucial for algorithm design and analysis. The use of symmetry simplifies and speeds up probabilistic analysis of stochastic algorithms. With the symmetry idea at its core, the suggested model effectively deals with random parameter probabilities, as shown in the development of game theory's utilization functions.

In addition to this introduction, the following sections are included: Section 2 discusses current state-of-the-art mobile database recovery strategies, Section 3 introduces the proposed mobile database recovery model, Section 4 describes the criteria for evaluating the proposed mobile database recovery based on game theory and presents the results, and Section 5 concludes and suggests directions for future work.

2. State-of-the-Art

In recent years, mobile application failure recovery has grown in popularity. The suggested solutions use checkpointing, logging, or a combination of the two, while taking into account the inherent constraints of the mobile computing environment. Much research is based on a distributed uncoordinated checkpointing method, in which several MHs may achieve a globally consistent checkpoint without depending on coordination messages. Others presented a checkpointing-only method, which offers globally consistent checkpoints without requiring additional communications but is unique in that it uses time to synchronize checkpoint creation.

The authors of [22] described a technique for mobile database applications to recover from checkpointing and logging failures depending on mobility. Current methods use periodic checkpoints that are not dependent on user mobility. This system initiates checkpoints only after a certain number of mobility handoffs has occurred. The failure, log arrival, and mobility rates of the mobile host determine the optimum threshold. This enables modification of the checkpointing rate on a per-user basis. Additionally, depending on the checkpoint frequency, the last checkpoint may be situated a considerable distance from the mobile support station (MSS). Additionally, a significant number of logs across several MSSs may be scattered, resulting in a lengthy recovery time.

The authors of [23] suggested a technique for recovering applications in a mobile computing environment by combining movement-based checkpoints with message recording. A node's adaptability is used to decide if a checkpoint should be taken. This method was developed using a variety of factors, including the number of MH registrations in an area, the number of regions, and the number of handoffs. This approach is especially beneficial in large networks with many areas. In contrast, operating in restricted areas may result in extra expenses.

The authors of [24] developed a rollback recovery method that prioritized separate checkpoints and message recording. The algorithm is unique in that it manages message logs and checkpoints through mobile agents. Additionally, if a mobile node travels a great distance from its most recent checkpoint, the agents are able to move the checkpoint and message logs stored in distant mobile service stations. Thus, the time needed to retrieve a mobile node would never exceed a specified threshold. It is feasible to keep just one checkpoint in permanent storage by recording messages. The main advantage of this study is the modest size of the message log, which cannot be very large owing to the network's low message substitution rate. Additionally, if a process interacts often, it may decrease its checkpointing interval. Nonetheless, this method occurs in a small number of situations, resulting in increased network activity during recovery. Specifically, if the length of the mobility profile exceeds the number of different mobile service stations at any point, the logs must be consolidated into a single place.

The authors in [25] prepared a proposal for a contemporary checkpointing method that is suitable for mobile computing systems. This method is characterized by its dependability and efficiency in terms of time-space overhead associated with checkpointing and normal application execution. The work presented in [26] suggested a log management and low-latency no-blocking checkpointing system that utilizes a mobile-agent-based architecture to reduce recovery time. By decreasing the amount of messages sent, this protocol reduced recovery time. On the contrary, particularly when many agents are needed, it may result in an increase in complexity, which may absorb some of the extra execution costs.

The authors of [27] developed a log management strategy for mobile computing systems that substantially lowers the total cost of failure recovery when compared to existing lazy and pessimistic approaches. Additionally, their approach enables recovery from a base station different than the one where it failed, lowering handoff costs, log replication costs, and the time required to recover from failure. The main benefit of their log management method is its ease of implementation, whereas the primary drawback is likely the recovery time if the home agent is situated a great distance from the mobile unit [28,29]. The authors of [30] described a recovery technique that is database and mobile device synchronization-dependent. As a consequence, the replication process guarantees that all organizations have consistent data. One drawback of this method is that, although it utilizes hash functions, it does not guarantee data integrity during transmission to the server, since both ends store the hash values in a database table.

The Need to Extend the Related Work

According to the review, the following are the current areas of research: (1) The majority of recovery studies employed a variety of techniques, including log management, checkpointing, movement-based checkpointing, and an agent-based logging scheme; (2) Because these techniques are so dissimilar, one cannot be used in place of another; this means that each algorithm has a distinct parameter set and different assumptions; (3) Despite the fact that some plans tried to merge several methods into a single contribution (hybrid method), they were damaged by the difficulties of selecting the optimal fusion from this pool of options. As a consequence, recovery costs may be high and the recovery mechanism may be excessively complicated; (4) The majority of schemes did not include environmental variables as influencing elements in the recovery process; and (5) As the demand for network applications grows, researchers are continuously developing new ways to solve the issue of high mobility or network connection loss owing to a variety of new or changing conditions. Thus, more fault-tolerant methods are needed to guarantee the continued functioning of mobile devices. As a consequence of the above, the use of recovery algorithms is constrained in a realistic manner. It is essential to design a plan that maximizes success via the selection of the most suitable recovery methods for the present situation. We selected quantum game theory over conflict analysis or interactive decision theory, as it enables us to compare the recovery possibilities available.

3. Quantum Game-Based Recovery Model

3.1. Mobile System Architecture

In a typical MDS design, a small database fragment is created from the main database on the MH. This design is meant to handle the accessibility limitations alleviated by MHs and mobile satellite services (MSS). If the MH is present in the cell serviced by the MSS, it may interact directly with another MH in the vicinity. The MH may freely move between cells, which each include a base station (BS) and a large number of MHs. Additionally, the BSs configured the stations to act as a wireless gateway, allowing them to communicate with the MHs and send data via the wireless network. Wireless communication is possible between the MHs and the BS, but not directly with the database server [11,31]. Figure 2 depicts the mobile system's architecture.

3.2. Recovery Modeling Using Quantum Game

The suggested method differs from prior MDS recovery attempts in that it takes into account a variety of important variables in the mobile environment during handoffs or service failures, which change depending on the situation, while conventional recovery algorithms are predicated on specific assumptions about the environment and operate accordingly.

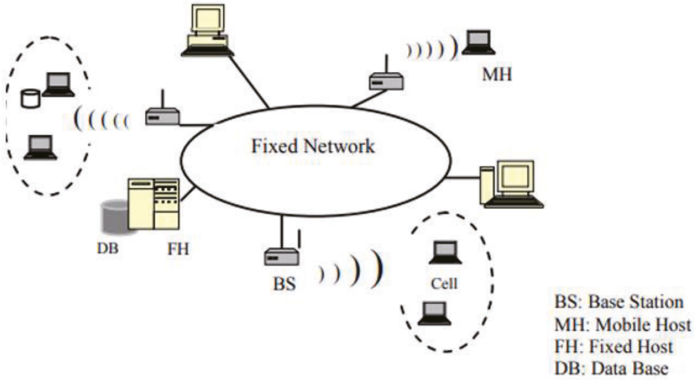


Figure 2. Mobile System Architecture.

The primary reasons for extending our prior work's two-player game to three players in this article are as follows: (1) reliance on a limited number of algorithms lowers the possibility of making good decisions, since it is possible for one of the algorithms to permanently dominate decision-making; (2) Allowing a greater number of algorithms to join the competition increases the efficacy of decision-making under a variety of environmental circumstances; (3) With a rise in the number of players, different degrees of complexity and mathematical calculations were used to address the recovery issue, resulting in enhanced capabilities for the proposed task; and (4) Several algorithms performed poorly in the present study, despite their success in earlier work. Thus, when the performance of certain algorithms deteriorated, they were eliminated from the competition; nevertheless, the entrance of others with superior results resulted in a substantial increase in performance, which is the purpose of presenting this study. The suggested model's architecture is shown in Figure 3. The following table (Table 1) summarizes the game assumptions utilized in the recovery modeling process.

Table 1. Game Assumptions.

Game	Parameter
Number of Players	Three Players
Game Type	Non Cooperative (No interaction between players)
Game Form	Strategic form
Evolutionary Game Theory	No
Strategy type	Pure not Mixed
Payoff Functions	TIME, Memory, Recovery Done Probability
The Winning Algorithm	The highest NE(Quantum Nash) in the reward matrix

Cooperative game theory (CGT) and non-cooperative game theory (NCGT) are two subfields of game theory. CGT elucidates how agents compete and cooperate to generate and capture value in unstructured interactions. NCGT simulates agents' activities, maximizing their usefulness based on a comprehensive description of each agent's motions and information. Cooperative games are ones in which players are convinced to follow a certain strategy via player dialogue and agreement. A strategy is a detailed plan of action that a player will follow in response to a variety of situations that may occur over the course of the game. On the other hand, non-cooperative games are ones in which participants select their own strategy of profit maximization. The main distinguishing feature is the absence of external authority to establish norms guaranteeing cooperative behavior. Without external authority (such as contract law), participants are unable to form coalitions and must compete alone [14,15].

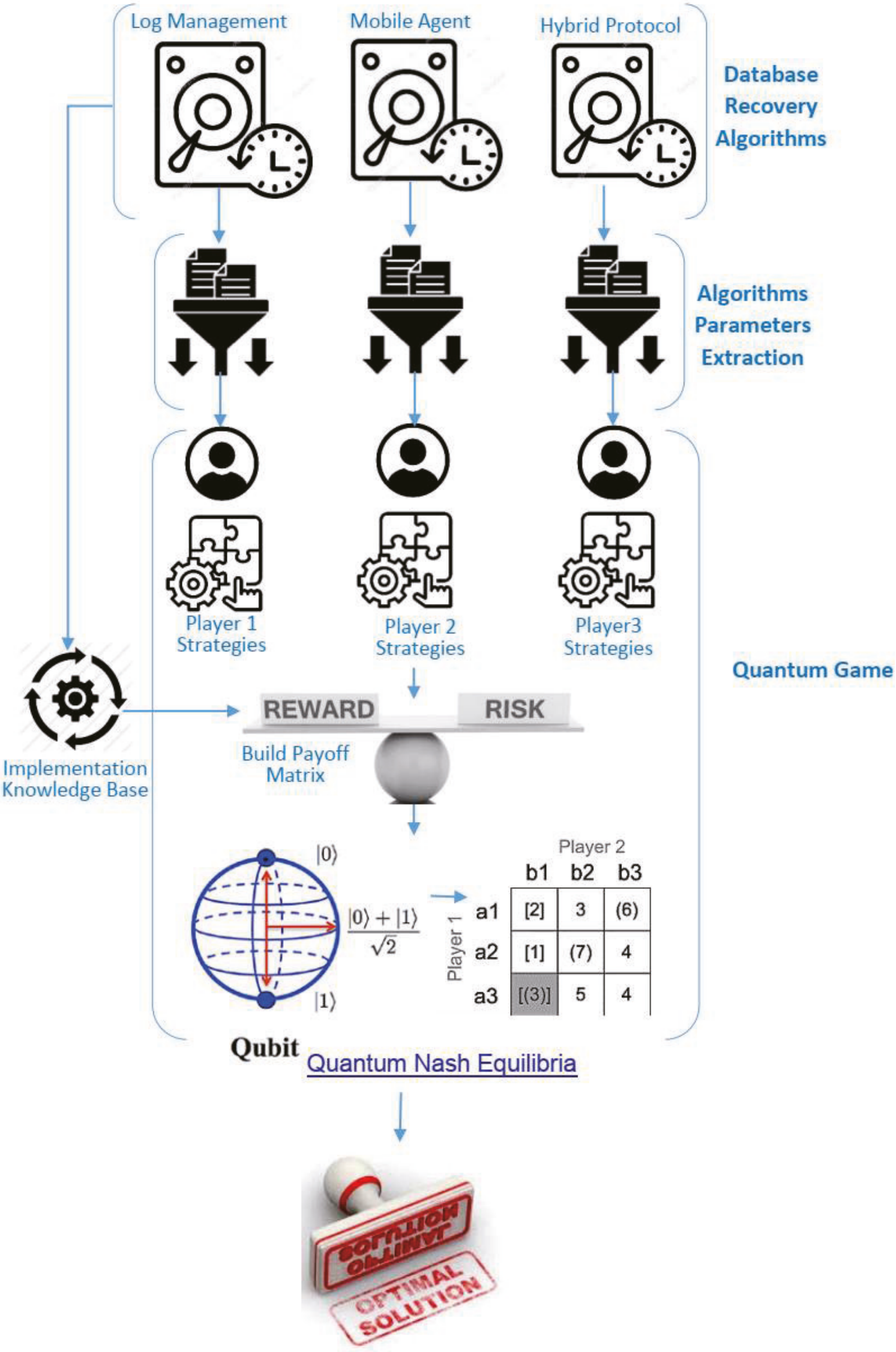


Figure 3. The Proposed MDS Recovery Model.

Non-cooperative games are often studied by trying to predict players' tactics and payoffs, as well as by finding Nash equilibria. Each player in NE is assumed to be aware of the other players' equilibrium tactics, and no one benefits from merely changing their strategy. If each player has chosen a strategy–a collection of actions based on previous game events—and no person can increase their expected payoff by changing their strategy while the other players retain theirs, then the current set of strategy choices characterizes NE.

The suggested game is modeled by static games with complete information, in which players simultaneously choose strategies and get rewards depending on the combination of actions taken. These types of games may be formalized using a normal-form representation [14]. This is a simple decision issue in which both players choose their actions concurrently (static game) and are rewarded for their mutual choices. Additionally, each player is fully aware of the values associated with his adversary's reward functions (complete information). The interrelationships of game theory are shown in Figure 4; for more information, see [14,15].

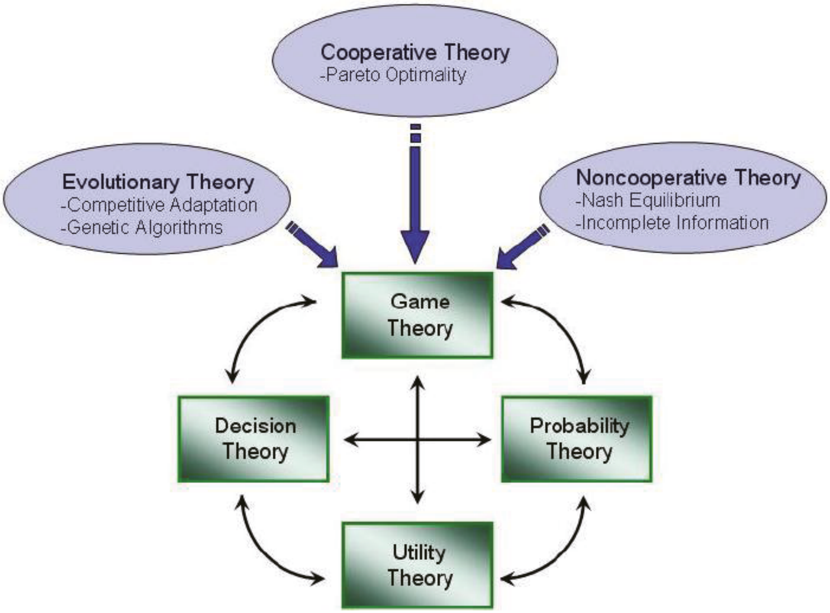


Figure 4. Game Theory Interrelationship Diagram.

3.2.1. Recovery Algorithms and Its Parameters

To demonstrate the technical importance of the system recovery model in the MDS, we evaluated the most widely used MD recovery algorithms in order to decide which methods should be explored further. We classified the recoverable algorithms in this situation according to their operation or features. As stated in [9–13], various groups differ in their approach to recovery. For our proposal, we selected three recovery protocols: log monitoring (as player 1) [27], mobile agent (as player 2) [24], and a hybrid method that combined movement-based check-pointing and message recording as a (as player 3) [22]. Because the real problem is not whether to adopt one of the well-known recovery techniques, but rather which strategy is most appropriate in light of the changes imposed by the operational environment, which is often unclear and changing. In this regard, the current research will ensure that the optimal recovery method is selected via the use of game theory and its essential variables.

To compete against other players, each player must create a set of strategies. To develop these methods, feature analysis and extraction are used for each chosen recovery procedure in order to determine the treatment's most strongest characteristics. Thus, in game theory, each selected protocol is defined by a player, and the way each protocol's variable is utilized determines the player's strategies. For instance, the first protocol (player 1) considered a variety of factors, including the log arrival rate, the handoff rate, the average log size, and the mobility rate. To summarize, the first player method involves retrieving the log file that was stored in the BS before to the failure, moving it to a new location linked to another BS, and updating the MH. The second protocol (player 2) makes extensive use of variables, including the number of processes in the checkpoint, a handoff threshold, and the length of the log. The second approach reduced recovery time by using a framework based on mobile agents. Here is a collection of processes in a list format. The home agent's list is included in the MH. The mobile agent traveled beside the MH and relayed information. The third process takes into account a variety of factors, including the total number of registrations in the area, the total number of regions, and the overall number of hand-offs. The work environment is split into several zones, and the checkpoint is only used once when MH enters and exits the region. For further information on how these protocols work, see [13–27].

It should be emphasized that any number of strategies for each player (protocol) may be produced by conducting any number of trials with various parameter values, although this increases the model's complexity. As a result, we believe it is important to choose several methods that reflect varying degrees of performance that may be depended upon in decision-making. To create the parameters for the necessary recovery algorithms using game theory as a decision-making method, we first apply the chosen protocols to the chosen key variables on which each protocol is reliant. Each algorithm is used for real database transactions in order to assess each player's strategy. A package is assessed that has an objective function for the total cost of recovery, which is computed differently for each method. In game theory, we build the payoff matrix for each protocol output value based on the previous stages. These outcomes are referred to as the utility or reward of each player. These payoffs or benefits are used to evaluate a player's level of satisfaction in a conflicting scenario.

In general, game theory may be summarized as follows: (1) a set of players (the negotiation algorithms chosen); (2) a pool of strategies for each player (the strategies take into account the assumed values of significant coefficients in each protocol, as well as possible environmental changes); and (3) the benefits or payoffs (utility) to any player for any possible list of the players' chosen strategies. It should be noted here that any number of strategies can be generated for each player (protocol) by running any number of experiments using different parameter values, but this of course increases the model's complexity. Therefore, we consider it best to choose a number of strategies to express different degrees of performance levels that can be relied upon in decision-making. In the suggested game-based recovery model, the assumption is that each player utilizes pure strategy and not a mixed one, as each strategy handles specific protocol parameters, and these parameters differ from one protocol to another. There are no general parameters used for all protocols. For pure strategies, it is far easier to obtain multiple solutions (of course if they exist) for the NE, and then select the best fitting one.

3.2.2. Build Knowledge Base

The suggested recovery model is predicated on the establishment of a knowledge base after the pre-implementation of each chosen recovery procedure in various simulated settings. The implementation knowledge base is created just once, and it is used to choose the optimal protocol based on the reward matrix and dominant equilibrium method. The decision is done here on the basis of the integration of three utility functions that serve as performance and evaluation benchmarks for the candidate protocols.

3.2.3. Build Payoff Matrix

In the proposed model's game, see Table 2, a finite collection of players $N = 3$ was used. Three players are allocated P_1, P_2, and P_3 in the three-player game. P_1 selects pure strategies from S_1, P_2 selects pure strategies from S_2, and P_3 selects pure strategies from S_3. If P_1 chooses the pure strategy i, P_2 chooses j, and P_3 chooses k, then the reward to P_1 is a_{ijk}, P_2 chooses b_{ijk}, and P_3 chooses c_{ijk}. Define the $S_1 \times S_2 \times S_3$ payoff "cubes" A, B, and C as [32,33]:

$$A = \left[a_{ijk}\right] \in R^{S1\times S2\times S3}, \mathcal{B} = \left[b_{ijk}\right] \in R^{S1\times S2\times S3},\ C = \left[c_{ijk}\right] \in R^{S1\times S2\times S3} \tag{1}$$

We create a matrix for each of player 3's actions (strategies); accordingly, player 1 selects a row, player 2 selects a column, and player 3 selects a table. Table 3 illustrates the bi-matrix for a three-player game with its payoff. As a result, each third player strategy k is represented separately in a matrix together with its associated reward in terms of player 1 and player 2's strategies. Here, a_{111} represents the payoff value for player 1's plan or strategy given the return function $u_1(s_1,\ t_1, c_1)$ if $(s_1,\ t_1, c_1)$ is selected, b_{111} denotes the payoff value for player 2's strategy provided the reward function $u_2(s_1,\ t_1, c_1)$ if $(s_1,\ t_1, c_1)$ is chosen, and c_{111} denotes the payoff value for player 3's strategy with payoff function $u_3(s_1,\ t_1, c_1)$. The proposed recovery model's game is a non-cooperative game in which all participants choose their own tactics in order to maximize their profit [15,32,33]. These kinds of games are amenable to formalization via the use of normal-form representations [34]. In a normal-form game, player i's strategy S_i'' strictly trumps player i's strategy S_i' if and only if:

$$U_i\ \left(S_i'',\ S_{-i}\right) \geq U_i\ \left(S_i',\ S_{-i}\right) \tag{2}$$

for every list of S_{-i} of the other players that represents all players' strategies except player i.

Table 2. The symbols used for description game theory model.

Symbol	Meaning
G	Quantum game model
N	Number of players
i	Game player
S	Player's strategy
U	Payoff function or utility
B_i	Best response for a player i

Table 3. The bi-matrix for three players' game.

Player	Player 3 Hybrid Method (Strategy k)				
	Player 2 Mobile Agent Method				
	Strategy	t_1	t_2	....	t_h
Player 1 Log Management Method	s_1	$(a_{111},\ b_{111}, c_{11k})$	$(a_{121},\ b_{121}, c_{12k})$	....	$(a_{1h1},\ b_{1h1}, c_{1hk})$
	s_2	$(a_{211},\ b_{211}, c_{21k})$	$(a_{221},\ b_{221}, c_{22k})$	....	$(a_{2h1},\ b_{2h1}, c_{2hk})$
		....	....	....	....
	s_m	$(a_{m11},\ b_{m11}, c_{m1k})$	$(a_{m21},\ b_{m21}, c_{m2k})$	....	$(a_{mh1},\ b_{mh1}, c_{mhk})$

Calculating the payoff in a game is complicated since it is dependent on the actions of other players. As a result, the strategy chosen by one player has an effect on the gain value of the other player. Three utility functions are included in this proposal as performance and assessment benchmarks for the candidate protocols. The functions are as follows: the amount of time consumed by the protocol during operation $(TIME_i)$ for each strategy, the amount of memory consumed during operation $(MEMO_i)$ for each strategy, and the rate expressing the percentage of recovery work completed ("recovery completion") for each strategy ($DONE\_PROB_i$). The point of this step is to determine the reward that the player will gain if its strategy wins according to the mobile environmental conditions. Since all players are assumed to be rational, they make their preferred decisions that maximize their rewards (payoff). Consequently, one player's strategy dominants another player's strategy if it always provides a greater payoff to that player regardless of the strategy played by the opposing player. Therefore, it is very important to determine the method of calculating the return for each player's strategy. Therefore, the aim of these functions is to evaluate every strategy by calculating an index (score) that represents its performance. Every function contains degrees to distinguish the better performance of each strategy with high degrees against the lower performance of all the utility functions.

After analyzing and executing the protocols, it was determined that each algorithm operates within a time range of 0 to 5 s, implying that the value of the return function from the time measurement would be distributed as follows:

$$if\begin{cases} C_{1,i} \in \,]0.1, 0] & u_i = 6 \\ C_{1,i} \in \,]0.5, 0.1] & u_i = 4 \\ C_{1,i} \in \,]0.9, 0.5] & u_i = 2 \\ C_{1,i} \in \,]1, 0.9] & u_i = 0 \\ C_{1,i} \in \,]2, 1] & u_i = -2 \\ C_{1,i} \in \,]5, 2] & u_i = -4 \end{cases}, \tag{3}$$

where $C_{1,i} = TIME_i$ is the time required to execute a strategy. $C_{2,i} = MEMO_i$, in the same context, is the amount of memory used by each protocol for a given strategy. The memory used during execution is expected to be between 0 and 4000 KB; therefore, the payoff values for the memory consumed by any strategy will be as follows:

$$if\begin{cases} C_{2,i} \in \,]500, 0] & u_i = 6 \\ C_{2,i} \in \,]1000, 500] & u_i = 4 \\ C_{2,i} \in \,]1500, 1000] & u_i = 2 \\ C_{2,i} \in \,]2000, 1500] & u_i = 0 \\ C_{2,i} \in \,]2500, 2000] & u_i = -2 \\ C_{2,i} \in \,]4000, 2500] & u_i = -4 \end{cases}, \tag{4}$$

Finally, when calculating the completion level of the recovery process, $C_{3,i} = DONE\_PROB_i$, where $DONE\_PROB_i$ is utilized to determine if recovery occurred in accordance with the handoff rates threshold. As a result, the value of the possible return measure for this work ranges from 0% to 100% and is distributed as follows:

$$if\begin{cases} C_{3,i} \in \,]20\%, 0] & u_i = 1 \\ C_{3,i} \in \,]40\%, 20\%] & u_i = 2 \\ C_{3,i} \in \,]60\%, 40\%] & u_i = 3 \\ C_{3,i} \in \,]80\%, 60\%] & u_i = 4 \\ C_{3,i} \in \,]100\%, 80\%] & u_i = 5 \end{cases} \tag{5}$$

3.2.4. Quantum Nash Equilibrium for Selection

Thus, the player's overall gain in this game is equal to the sum of the reward values associated with the variables ($C_{1,i}$, $C_{2,i}$, $C_{3,i}$). The ultimate solution may be obtained in one of two ways: (1) by achieving a single and exclusive dominant equilibrium method in the game, or (2) by using NE [15]. The strategies produced via the first method, dubbed iterated elimination of strictly dominated strategies, reflect the optimal actions that each player might rationally take, and therefore comprise the game's (rational) solution. Regrettably, this alluring approach yields no prediction at all for some kinds of situations in which no strategy survives the elimination phase. In this situation, it is unclear which course of action would be deemed reasonable and best. This is a unique game in which there are no absolutely dominant strategies.

NE is a game theory concept that determines the optimal result in a non-cooperative game in which each player has little incentive to change his or her initial strategy. Under the NE, a player wins nothing by departing from their initial strategy, guaranteeing that the strategies of the other players stay constant as well. A game may include several NE states or none at all [35]. Unfortunately, the majority of games lack dominating strategies. Thus, if there are many solutions (more than one NE) to a given issue, the alternative is to find another handling mechanism. To address this issue, all values in the payoff matrix are subject to an addition or subtraction mechanism based on one of the critical factors, such as execution time, so that more points may be awarded to the quickest element and vice versa (normalization and reduction phase). Finally, the updated payoff matrix is utilized to identify a more optimal solution (Pure Nash) that matches the various environmental factors.

Since all players are assumed to be rational, they make their preferred decisions which maximize their rewards. Consequently, one strategy for a player is dominant over another strategy for another player if it always provides a greater payoff to that player regardless of the strategy played by the opposing player. Therefore, it is very important to determine the method of calculating the return for each strategy for each player. A particular algorithm is selected when its strategy achieves the highest NE in the reward matrix.

In contrast to the classical situation, where the theory is incapable of making any unique prediction, the application of quantum formalism will show a new property: the emergence in entangled strategies of a NE reflecting the unique solution to the game. In the quantum version of this three-player game, players execute their strategies by applying the identity operators they possess with probability p, q, and r to the starting quantum state, respectively. The three players apply the inversion operator σ with probability (1-p), (1-q), and (1-r). If ρ_{in} in is the density matrix corresponding to the initial quantum state, then the final state after players have implemented their strategies is [17,18].

$$\rho_{fin} = \sum_{U=I,\sigma} P(H_A)P(H_B)P(H_C)H_A \otimes H_B \otimes H_C\, \rho_{in}\, H_A^{\dagger} \otimes\ H_B^{\dagger}\ \otimes\ H_C^{\dagger} \tag{6}$$

where either I or σ may be used as the unitary and Hermitian operator H. $P(H_A)$, $P(H_B)$, and $P(H_C)$ are the probability that players A, B, and C, respectively, will apply the operator H to the initial state. ρ_{in} is a convex combination of all quantum processes. Assume the arbitrator creates the following pure initial quantum state with three qubits (two strategies for each player for simplicity):

$$\begin{gathered} |\psi_{in}\rangle = \sum_{i,j,k=1,2} C_{ijk}|ijk\rangle \\ \sum_{i,j,k=1,2} \left|C_{ijk}\middle|ijk\right|^2 = 1 \end{gathered} \tag{7}$$

where the quantum state's eight basis vectors are $|ijk\rangle$ for i, j, and k equal to one and two. The starting state may be thought of as a global state (in a (2 $\otimes$ 2 $\otimes$ 2)-dimensional Hilbert space) of three quantum two-state systems or 'qubits'. The unitary operators I and σ are used by the player with conventional probability ρ_{in} included into its strategic.

As a consequence, rather of considering just a discrete and finite set of strategies, we will now consider their linear superposition by endowing the strategic space with the formal structure of a Hilbert space. As a result, pure quantum strategies may be constructed, which are characterized as linear combinations of pure classical strategies with complex coefficients. This must be interpreted as the probability of using a single pure classical method. It is worth noting that this interpretation of pure quantum strategies is identical to the classical concept of a mixed strategy introduced previously, because we are currently considering a restricted class of games (static games), which lacks typical quantum interference effects between amplitudes [36,37].

4. Discussion

4.1. Simulation Setup

The simulation is designed to evaluate the proposed model for MD recovery based on game theory. In this respect, we implemented the prototype NS2 program using two software packages: Matlab and NS2. By modeling discrete occurrences, the NS2 simulation software enables developers to improve their businesses in real time. Additionally, it supports a variety of protocols such as TCP, routing, and multicast across wired and wireless networks and runs on a variety of platforms including Linux and Windows [38,39]. As a result, we use this software to implement the stage of collecting data on the work environment at various levels in order to mimic changes in the work environment. Then, Matlab software is used to construct a game theory in order to assess various recovery procedures by inputting the output values from the simulation stage in order to find the optimal decision. For our solution, we used mobile log files of various sizes that included the process data that each method would obtain.

The settings for the NS2 simulation are summarized in Table 4. The MAC layer protocol of IEEE 802.11 for wireless large area networks is utilized here. A movement file provides the mobile client node's motions. The mobile client node transmission range is 250 m. Each cell has one base station. A random waypoint (RWP) model determines the starting node position and movement. The RWP model is based on random locations, speeds, and halt durations. The prototype was built in modules and tested on a Dell Inspiron N5110 laptop from Dell Computer Corporation in Texas. Processor: Intel(R) Core(TM) i5–2410M, 4.00 GB RAM, Windows 7 64–bit. The proposed model's efficiency is assessed using execution time and recovery probability. See [24,27] for more details. The results are the average of several repeated experiments for different initial location and movements of the nodes due to the using of RWP model. The most suitable values for the collection of protocol's factors (strategies) were picked from the literature, based on the assessment of selected recovery protocols, to represent the protocol's performance in a variety of settings.

4.2. Simulation Results

4.2.1. Experiment One

Aim: The first set of experiments was designed to evaluate the performance of the suggested recovery model in terms of actual execution time as a function of log file size. In general, the cost of recovery is very low for any scheme, since the full log information is stored at the current base station. When the MH travels a great distance from the initial BS, the difficulty of the recovery method used to locate and transmit the log file rises. Increasing the file size results in an increase in the transmission cost.

Main Results: A As shown in Table 5, the suggested game theory-based recovery model delivers a faster execution time while increasing file size in a variety of mobile settings, depending on the simulation's changing nature.

Discussion: One reason for these results is that, since the proposed model is based on a knowledge base that was developed after the pre-implementation of each chosen recovery protocol in various simulated settings, it chooses the best appropriate recovery protocol for the present circumstance (variation of log file size). As a result, the proposed model requires less time to execute for numerous simulation runs.

4.2.2. Experiment Two

Aim: The second set of experiments evaluated the suggested recovery model's performance as a function of mobility rate in terms of actual execution time. The suggested recovery model is based on the creation of a knowledge base that is created once and used to choose the best suitable procedure based on the reward matrix and dominant equilibrium method. The decision is done here on the basis of the integration of three utility functions that serve as performance and evaluation benchmarks for the candidate protocols.

Table 4. Simulation Parameters.

Variable	Meaning
Channel type	Channel/Wireless Channel
MAC type	Mac/802_11
Radio-propagation model	Propagation/Two Ray Ground
Network interface type	Phy/wireless Phy
Interface queue type	queue/drop tail/priqueue
Antenna model	Antenna/Omni Antenna
Link layer type	LL
Routing protocol	Destination-Sequenced Distance Vector (DSDV)
Coordinate of topology	670 m × 670 m
Max packet in *ifq*	500
Time to stop simulation	250

Table 5. Execution time (Seconds) as a function of log file size.

Algorithm/Log Size	5 KB	30 KB	60 KB	90 KB	120 KB	150 KB
Log Management	1	1.3	1.8	2.5	3.3	3.5
Mobile Agent	1.1	1.5	1.9	2.4	3	3.15
Hybrid Method	1.2	1.6	1.7	2.3	2.8	3
Proposed Model	1	1.4	1.8	2.3	2.7	2.75

Main Results: Figure 5 shows that when mobility rate rises, recovery cost increases. In addition, the current base station has all the log information, lowering the recovery cost. The recovery cost is greater when the mobile node is recovered in the same base station.

Discussion: The results in Figure 5 show that despite their success in limited regions, log management and agent-based methods usually suffer from increased implementation time in the long term. On the contrary, the hybrid approach may be more successful in a vast environment than in a small area since it only requires a checkpoint once before MH moves across regions, thus saving time.

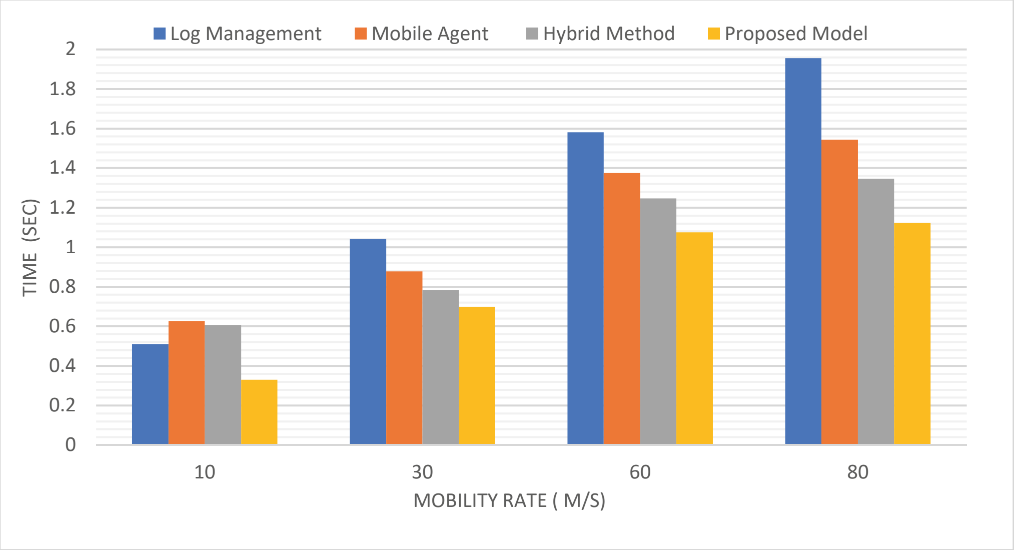


Figure 5. Execution time as a function of mobility rate.

4.2.3. Experiment Three

Aim: The final set of experiments examined the connection between the likelihood of recovery and the handoff threshold. There is a favorable correlation between increasing the handoff rate and the likelihood of completing the recovery process. Handoff involves moving a MS from one base station to another. Algorithms with set parameters do not work well in varying system configurations. Handoff algorithms should take into consideration the communication system's peculiarities. If the preparation time of rapid handoff is longer than the WLAN sojourn time linked to mobile node speed, the handoff fails and packets are lost. If the mobile node speed is too slow, handoffs are initiated too late, reducing WLAN service duration. The handoff cost comprises the checkpoint status, message log, and acknowledgement.

Main Results: Figure 6 plots the completion of the recovery process against increases in the hand-off threshold rate. The recovery likelihood is significantly reduced when using the log management technique, but the hand-off threshold rate is raised. Alternatively, some techniques decreased gradually when the threshold value was increased.

Discussion: The results demonstrated that the log technique is applicable only in small work settings. Whereas the hybrid approach and the agent-based method both performed well for areas with multiple regions or regions located farther from the site of retrieval. As anticipated, the suggested model has a higher recovery probability in the long term when compared to the other methods in the various simulated settings.

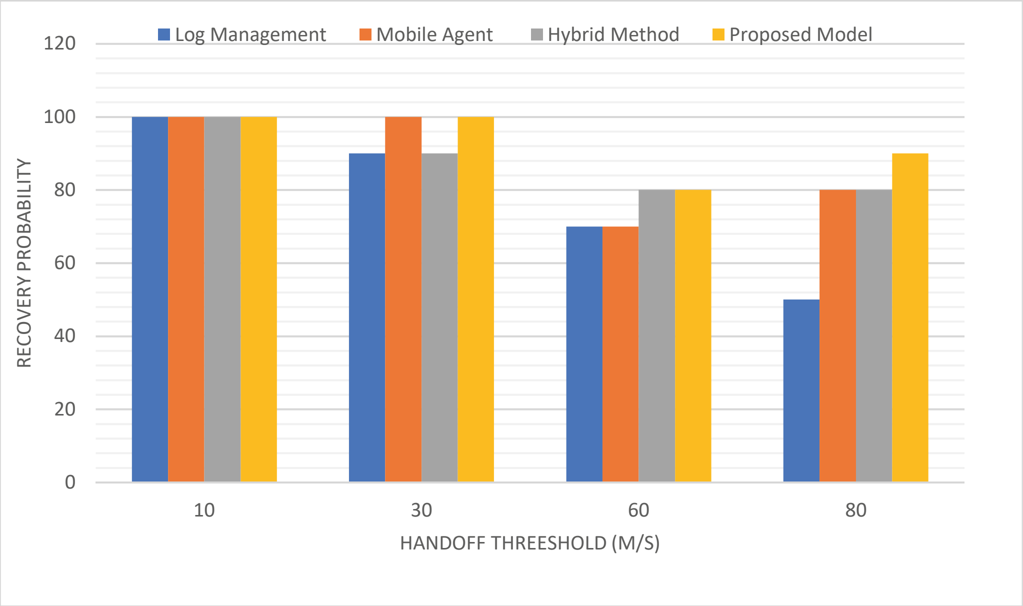


Figure 6. Recovery probability vs. handoff threshold.

4.2.4. Experiment Four

Aim: The next set of experiments was conducted to measure the efficiency of the proposed model compared to traditional recovery methods concerning the degree of complexity of the chosen strategies. In this case, the degree of complexity of the strategies was divided into three levels: low, medium, and high. The difficulty here is measured through utility functions. In the case of the log file size parameter, the range of size changes from small to medium to large is a measure of the complexity of the strategy. The same is the case for the other parameters such as handoff rate and mobility rate. Herein, the evaluation is based on the total payoff value that is calculated as the sum of the three utility functions' outputs.

Main Results: The results in Figure 7 confirm the superiority of the suggested model. As previously stated, since the proposed model is based on a knowledge base that was created after the pre-implementation of each chosen recovery protocol in various simulated environments, it automatically picks the most appropriate recovery procedure for the present situation.

Discussion: As expected, the log management algorithm is preferred in the small area. However, this decision turns out to be unfavorable with a large log size, especially in distant regions, because the cost of transfer the log file becomes high, and thus the cost of recovery becomes more problematic. On the other hand, the hybrid method gave a good payoff compared to other algorithms whenever there was a multiplicity in the regions because it had taken the recovery point once. The same is true for agent-based recovery, as it is easy to find the recovery location by tracing the MH ids.

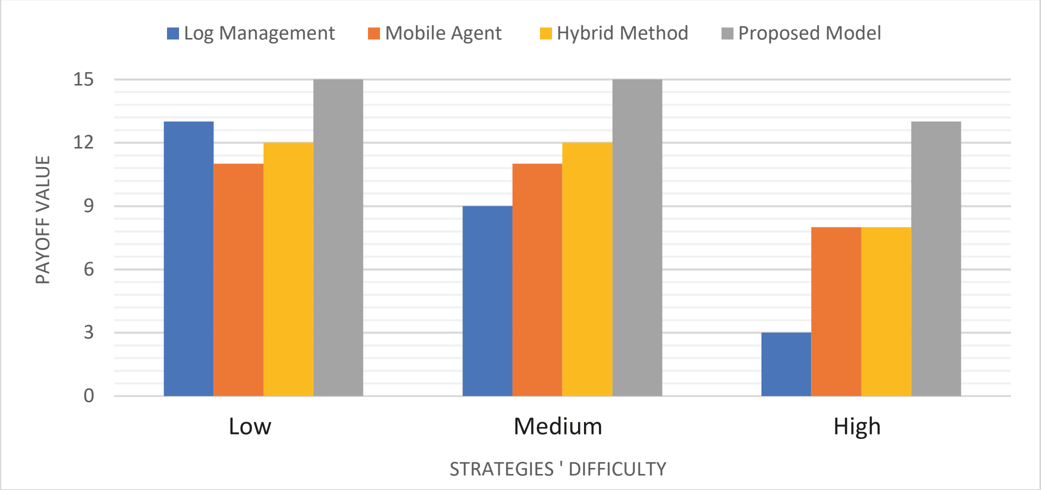


Figure 7. Comparison results in terms of total payoff.

4.2.5. Experiment Five

Aim: A comparative analysis has been done with an optimization model dealing with the recovery of HLR mobility databases presented in [40]. In this work, analytical performance results evaluation for the location updating scheme have been done for the mobility database failure recovery under some assumptions. The lost incoming calls cost is calculated versus the system probability distribution for the recovery time. Therefore, in our proposal, a simulation for a traffic analysis with random log file sizes is done into failure recovery time to calculate the lost packet rates.

Main Results: The results in Figure 8 show a significant decrease in the average lost packets in the proposed model compared to the comparative protocol under different failure recovery time.

Discussion: That is because the proposed model switched between a pool of recovery algorithms that may suitable for any environment rather than using a fixed recovery method that may not be suitable, which leads to the retrieval system not constantly updating its database in a proper time. Thus, it reduces the packet loss rate.

4.2.6. Experiment Six

Aim: The last set of experiments was implemented to compare the suggested recovery model (the extended version with three players) with our previous version (modelling by two players). The precision (accuracy) rate has been used for assessment according to the recovery probabilities for multi degrees of handoff threshold values that are considered as a very influential factor to complete the retrieval process.

Main Results: As expected, Form Figure 9, utilizing three players for game modelling, increases the precision rate by an average of 5% compared with two players for a game.

Discussion: This improvement comes from increasing competition through utilizing more than two recovery algorithms that gave the opportunity to select the best protocol from a pool of algorithms according to the current MDS environmental conditions. Therefore, some protocols used in our previous work [21] will be retreated in this work due to the entry of other protocols in the competition.

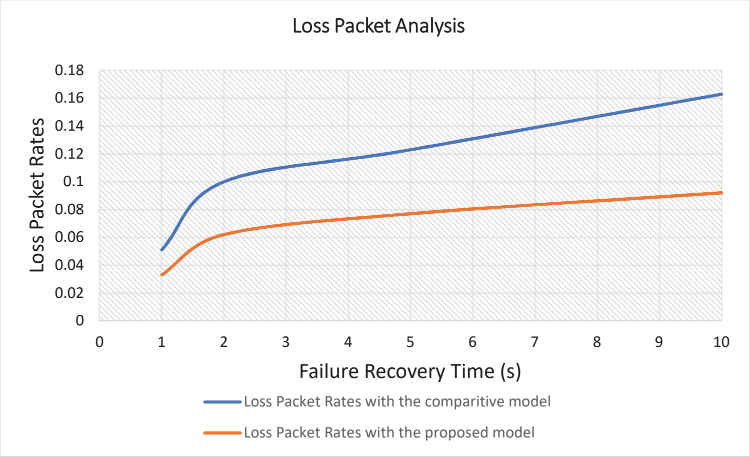


Figure 8. A loss packet analysis with a comparative model.

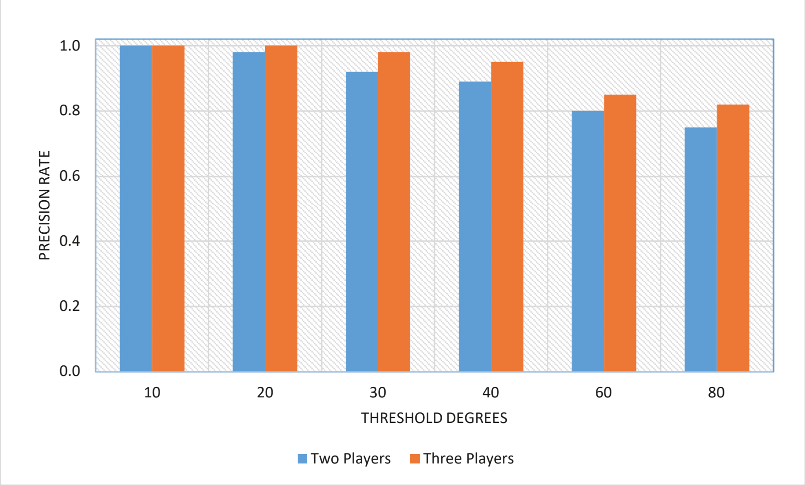


Figure 9. Accuracy analysis between two-player- and three-player-based game modeling.

5. Conclusions and Future Work

The purpose of this paper was to propose a new game theory model for determining the optimal recovery strategy in MSD. The novel method was compared against three of the most commonly used MDS recovery procedures in a competitive environment. Game theory is founded on the idea that each algorithm chooses the most appropriate strategy in terms of message delivery time and message count in order to determine the right recovery solution based on environmental factors. A key step of a quantum game-theoretic research is identifying which strategy to a recovery process is the superior solution to the strategies chosen by others. The proposed recovery model is based on the development of a knowledge base that is used to choose the best appropriate method based on the reward matrix and dominant equilibrium technique. The experimental findings demonstrate the superiority of the suggested recovery paradigm. In the future, it may be essential to include more recovery procedures to optimize the suggested model's performance. Additionally, a hybrid method based on game theory and the recently developed paradigm of cloud algorithms was utilized to improve the outcome. Furthermore, this enhanced the game model to allow interactions between players and utilized mixed strategies. Prior to that, some investigations should be made into the probability distribution of the behavior of the competing players (dealing with uncertainty). The application of the proposed model in the case of large systems and the discussion of its complexity will be also done in future work. Finally, the concept of mind-light-matter unity AI/QI in quantum-inspired computing can be utilized to enhance the suggested model.

Author Contributions: Conceptualization, M.M.M., S.M.D. and Y.F.M.; methodology, S.M.D. and Y.F.M.; software, Y.F.M.; validation, M.M.M. and S.M.D.; formal analysis, M.M.M., S.M.D. and Y.F.M.; investigation, M.M.M. and S.M.D.; resources, S.M.D. and Y.F.M.; data curation, S.M.D. and Y.F.M.; writing—original draft preparation, M.M.M., S.M.D. and Y.F.M.; writing—review and editing, M.M.M. and S.M.D.; visualization, Y.F.M.; supervision, M.M.M. and S.M.D.; project administration, S.M.D. and Y.F.M.; funding acquisition, Y.F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rathika, R.; Prakash, T.; Aloysius, A. A survey on mobile computing security. *Int. J. Res. Anal. Rev.* **2019**, *6*, 645–649.
2. Anandhakumar, M.; Ramaraj, E.; Venkatesan, N. Query issues in mobile database systems. *Asia Pac. J. Res.* **2014**, *1*, 24–36.
3. Ibikunle, F.; Adegbenjo, A. Management issues and challenges in mobile database system. *Int. J. Eng. Sci. Emerg. Technol.* **2013**, *5*, 1–6.
4. Su, H. The processing technology in mobile database transaction system. *Int. J. Database Theory Appl.* **2015**, *8*, 51–60. [CrossRef]
5. Patil, P. Mobile computing: Issues and limitations. *Int. J. Comput. Sci. Mob. Appl.* **2016**, *4*, 1–6.
6. Nisar, J.; Trumboo, O.N. Database for mobile application. *Int. J. Trend Sci. Res. Dev.* **2018**, *2*, 853–854. [CrossRef]
7. Mobarek, A.; Abdelrhman, S.; Abdel-Mutal, A.; Adam, S.; Elbadri, N.; Ahmed, T.M. Transaction processing, techniques in mobile database: An overview. *Int. J. Comput. Sci. Appl.* **2015**, *5*, 1–10. [CrossRef]
8. Selvarani, D.; Ravi, T. A survey on data and transaction management in mobile databases. *Int. J. Database Manag. Syst.* **2012**, *4*, 1211. [CrossRef]
9. Ramaj, V.; Januzaj, Y.; Januzaj, Y. Transactions management in mobile database. *Int. J. Comput. Sci. Issues* **2016**, *13*, 39–44.
10. Ahmed, I.; Karvonen, H.; Kumpuniemi, T.; Katz, M. Wireless communications for the hospital of the future: Requirements challenges and solutions. *Int. J. Wirel. Inf. Netw.* **2020**, *27*, 4–17. [CrossRef]
11. Kumar, P. Fault tolerance approach in mobile distributed systems. *Int. J. Comput. Appl.* **2015**, *975*, 8887.
12. Chandna, A. Mobile database design: A key factor in mobile computing. In Proceedings of the 5th National Conference on Computing for Nation Development, New Delhi, India, 10–11 March 2011; pp. 1–4.

13. Kumar, V. Mobile database recovery. In *Fundamentals of Pervasive Information Management Systems*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2013; pp. 219–238.
14. Murthy, D.; Jack, N. Game theoretic modelling of service agent warranty fraud. *J. Oper. Res. Soc.* **2017**, *68*, 1399–1408. [CrossRef]
15. Bonau, S. A Case for Behavioral Game Theory. *J. Game Theory* **2017**, *6*, 7–14.
16. Quezada, L.; Dong, S. Quantum version of a generalized Monty Hall game and its possible applications to quantum secure communications. *Ann. Physik* **2021**, *533*, 2000427. [CrossRef]
17. Zabaleta, O.; Barrangú, J.; Arizmendi, C. Quantum game application to spectrum scarcity problems. *Phys. A Stat. Mech. Its Appl.* **2017**, *15*, 455–461. [CrossRef]
18. Babu, S.; Mohan, U.; Arthanari, T. Modeling coopetition as a quantum game. *Int. Game Theory Rev.* **2020**, *22*, 2040001. [CrossRef]
19. Schmid, C.; Flitney, A.P.; Wieczorek, W.; Kiesel, N.; Weinfurter, H.; Hollenberg, L.C. Experimental implementation of a four-player quantum game. *New J. Phys.* **2010**, *12*, 063031. [CrossRef]
20. Satria, D.; Park, D.; Jo, M. Recovery for overloaded mobile edge computing. *Future Gener. Comput. Syst.* **2017**, *70*, 138–147. [CrossRef]
21. Mokhtar, Y.F.; Darwish, S.M.; Madbouly, M.M. An enhanced database recovery model based on game theory for mobile applications. *Adv. Intell. Syst. Comput.* **2021**, *1261*, 16–25.
22. Saraswat, B.K.; Suryavanshi, R.; Yadav, D.S. A comparative study of checkpointing algorithms for distributed systems. *Int. J. Pure Appl. Math.* **2018**, *118*, 1595–1603.
23. Silberschatz, A.; Korth, H.F.; Sudarshan, S. Recovery system. In *Database System Concepts*, 7th ed.; McGraw-Hill Education: New York, NY, USA, 2020; pp. 907–957.
24. George, S.E.; Chen, I.R. Movement-based checkpointing and logging for failure recovery of database applications in mobile environments. *Distrib. Parallel Databases* **2008**, *23*, 189–205. [CrossRef]
25. Jaggi, P.; Singh, A. Log based recovery with low overhead for large mobile computing systems. *J. Inf. Sci. Eng.* **2013**, *29*, 969–984.
26. Chowdhury, C.; Neogy, S. Checkpointing using mobile agents for mobile computing system. *Int. J. Recent Trends Eng.* **2009**, *1*, 26–29.
27. Mansouri, H.; Badache, N.; Aliouat, M.; Pathan, A. A new efficient checkpointing algorithm for distributed mobile computing. *Control. Eng. Appl. Inform.* **2015**, *17*, 43–54.
28. Mahmoodi, M.; Baraani, A.; Khayyambashi, M. Recovery time improvement in the mobile database systems. In Proceedings of the International Conference on Signal Processing Systems, Singapore, 15–17 May 2009; pp. 688–692.
29. Pamila, M.J.; Thanushkodi, K. Log management support for recovery in mobile computing environment. *arXiv* **2009**, arXiv:0908.0076.
30. Chowhan, R.S. Mobile agent programming paradigm and its application scenarios. *Int. J. Curr. Microbiol. Appl. Sci.* **2018**, *7*, 3269–3273. [CrossRef]
31. Belghiat, A.; Kerkouche, E.; Chaoui, A.; Beldjehem, M. Mobile agent-based software systems modeling approaches: A comparative study. *J. Comput. Inf. Technol.* **2016**, *24*, 149–163. [CrossRef]
32. Kekgathetse, M.; Letsholo, K. A survey on database synchronization algorithms for mobile device. *J. Theor. Appl. Inf. Technol.* **2016**, *86*, 1–9.
33. Bhagat, A.R.; Bhagat, V.B. Mobile database review and security aspects. *Int. J. Comput. Sci. Mob. Comput.* **2014**, *3*, 1174–1182.
34. Liang, X.; Yan, Z. A survey on game theoretical methods in human–machine networks. *Future Gener. Comput. Syst.* **2019**, *92*, 674–693. [CrossRef]
35. Shaha, H.; Kakkada, V.; Patela, R.; Doshi, N. A survey on game theoretic approaches for privacy preservation in data mining and network security. *Procedia Comput. Sci.* **2019**, *155*, 686–691. [CrossRef]
36. Orbay, B.Z.; Sevgi, L. Game theory and engineering applications. *IEEE Antennas Propag. Mag.* **2014**, *56*, 256–267.
37. Gerasimou, G. Dominance-solvable multi-criteria games with incomplete preferences. *Econ. Theory Bull.* **2019**, *7*, 165–171. [CrossRef]
38. Trivedi, H.; Mali, M. A Review on Network Simulator and its Installation. *Int. J. Res. Sci. Innov.* **2014**, *1*, 115–116.
39. Gayathri, C.; Vadivel, R. An overview: Basic concept of network simulation tools. *Int. J. Adv. Res. Comput. Commun. Eng.* **2017**, *6*, 19–22.
40. Fang, Y.; Chlamtac, I.; Fei, H. Failure recovery of HLR mobility databases and parameter optimization for PCS networks. *J. Parallel Distrib. Comput.* **2000**, *60*, 431–450. [CrossRef]

symmetry

MDPI

Article

Information Hiding Based on Two-Level Mechanism and Look-Up Table Approach

Jeng-Shyang Pan [1], Xiao-Xue Sun [1], Hongmei Yang [1], Václav Snášel [2] and Shu-Chuan Chu [1,*]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China; jengshyangpan@gmail.com (J.-S.P.); xues1123@163.com (X.-X.S.); yanghongmei@sdust.edu.cn (H.Y.)
[2] Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, 70032 Ostrava, Czech Republic; vaclav.snasel@vsb.cz
* Correspondence: scchu0803@gmail.com

Abstract: Information hiding can be seen everywhere in our daily life, and this technology improves the security of information. The requirements for information security are becoming higher and higher. The coverless information hiding with the help of mapping relationship has high capacity, but there is still a problem in which the secret message cannot find the mapping relationship and the process requires extra storage burden during the transmission. Therefore, on the basis of symmetric reversible watermarking, the paper introduces the two-level mechanism and novel arrangements to solve the problem of sufficient diversity of features and has better capacity and image quality as a whole. Besides, for the security of secret message, this paper designs a new encryption model based on Logistic mapping. This method only employs coverless information hiding of one carrier image to transmit secret message with the help of the two-level mechanism and look-up table. Reversible information hiding is applied to embed the generated location table on the original image so that ensures storage and security. The experiment certifies that the diversity of hash code is increased by using the two-level image mechanism and the quality of the image is excellent, which proves the advantages of the proposed symmetric method over the previous algorithm.

Keywords: logistic mapping; two-level mechanism; look-up table; reversible information hiding

Citation: Pan, J.-S.; Sun, X.-X.; Yang, H.; Snášel, V.; Chu, S.-C. Information Hiding Based on Two-Level Mechanism and Look-Up Table Approach. *Symmetry* **2022**, *14*, 315. https://doi.org/10.3390/sym14020315

Academic Editor: José Carlos R. Alcantud

Received: 2 January 2022
Accepted: 28 January 2022
Published: 3 February 2022

1. Introduction

The development of network technology has greatly promoted the security need in daily life [1,2]. Our daily life is inseparable from the application of network transmission, so there are many technologies used to solve the problem of network transmission security. The digital media transmitted by the network mainly includes video, audio, pictures, and text. As the main form of multimedia, the acquisition and tampering of digital images are quite easy. At present, the most commonly used methods to protect digital media are mainly grouped by encryption and information hiding. The former is that the sender applies the encryption algorithm to directly encrypt the multimedia information, and the receiver with the key can obtain the secret information in view of the encryption method. Generally speaking, the garbled state formed after encryption can easily arouse the suspicion and attention of others. The information hiding needs to conceal the existence of the secret message with the help of the carrier, that is, it is hidden in multimedia data. The secret is transmitted depending on multimedia data. The receiver exploits the corresponding extraction method to extract the secret message.

The information hiding technology is a key technology and is widely utilized in many fields such as copyright protection and digital signature [3,4]. Information hiding technology changes the carrier to hide information, which can be implemented in the spatial domain or frequency domain. Some techniques even permanently damage the original carrier image. Sahu et al. apply a two-level LSB replacement technique [5].

Muhuri et al. implement image steganography on Integer Wavelet Transformation (IWT) using Particle Swarm Optimization (PSO) algorithm in 2020 [6–8]. Digital watermarking focuses on protecting the secret message. The digital watermarking methods commonly used inevitably cause irreversible effects on the original carrier [9]. Therefore, the emergence of reversible information hiding technology satisfies the dual recovery of the carrier and secret message [10–12]. Reversible information hiding means that when the receiver receives the watermarked image not only the secret message can be extracted but also the original image can be recovered according to the embedding and extraction rules [13,14]. It not only satisfies the confidentiality of secrets but also does not permanently destroy the image, which can be widely applied to medical image transmission.

Reversible information hiding was first proposed by Honsinger et al. in 1999 [15]. There are three types of algorithms on the spatial domain most commonly used by reversible information hiding techniques: lossless compression, difference expansion [16], and histogram shifting [17,18]. Difference expansion uses the correlation characteristics between pixels to expand the pixel difference through integer transformation to embed secret [19,20]. The prediction error algorithm predicts the target pixel through the prediction model [21–23]. Weng et al. combine difference expansion and prediction error to improve the quality of the image embedded with data [24]. According to whether the parameters used for embedding and extraction are the same, the information hiding can be divided into symmetric and asymmetric. The symmetric mechanism is efficient and simple so most of the current information hiding technologies are symmetric [25–27].

Steganalysis is a technique that tries to find the secret message. Although the secret message is difficult to be found by Human Visual System (HVS) on the carrier, the existence of the secret message can still be found through the traces of these modifications [28]. Coverless information hiding is one of the means to avoid steganalysis. This method directly expresses the secret message utilizing the characteristics of the carrier [29]. Zhou et al. use faster-RCNN for training to find the labels of images to express the secret message [30]. The network architecture based on deep learning has a large transmission overhead and requires a long training time. Zou et al. come up with a novel coverless information hiding based on the average pixel value of the image, which hides the information through mapping relation and multi-level index structure [31]. Cao et al. propose an approach based on the molecular structure images of material [32]. Wang et al. construct an intelligent search algorithm for mapping relationships to implement coverless information hiding [33]. Although this method solves the problems of transmission overhead and training, it inevitably adopts the mapping relationship. Information hiding techniques based on mapping relationships still have the limitation that increases the number of mapping relationships with the extension of secret information, resulting in a large cost or even being impractical.

The coverless information hiding approach proposed by Fatimah Shamsulddin Abdulsattar well improves the efficiency of feature extraction and explores the effect of block size on image hiding [34]. However, the features generated by analysis may not satisfy the secret message embedding due to lacking diversity in the obtained hash code. To improve the success rate of secret hiding, we combine the two-level mechanism and design novel arrangements to increase the diversity of hash codes.

During the hiding process, a location table is generated, so additional storage space and transmission process are required to ensure the recovery of secret data. The hidden framework proposed in this paper combines coverless information hiding and reversible information hiding techniques [35]. This process not only satisfies the hiding capacity of the secret message but also ensures the recovery of the original image. To better solve the problem of additional information storage and the security of secret data, the newly proposed encryption model is employed to encrypt the data first, then generate eigenvalues, calculate the hash code, and establish a look-up table on the original image [36]. The generated location table is embedded taking advantage of reversible information hiding technology, and the whole process is symmetric.

Since the secret message is hidden adopting a coverless way, this method will not produce any changes to the image, and then the pivotal information is embedded using the prediction error expansion (PEE) algorithm. Of course, other outstanding PEE algorithms could also be combined. Finally, the image can be recovered. The main contributions of this work are as follows.

1. A more secure encryption model based on Logistic mapping is devised;
2. The hash code based on the two-level image mechanism is more diverse and reduces the unconcealable rate of the secret message;
3. A new combination of reversible information hiding and coverless information hiding is designed, which greatly improves capacity and the image quality;
4. The proposed method solves the additional storage of location table without sacrificing hiding capacity and no large image database is required;
5. Compared with other similar algorithms, our method has more security and better image quality and higher storage capacity.

The remainder of the paper is formed as follows. Section 2 reviews the basic Logistic mapping and previous coverless information method. The proposed model is presented in Section 3. Section 4 displays experiment results and comparative analysis. Finally, Section 5 gives a conclusion and future directions.

2. Related Works

To further improve the security of secret message, the paper designs a novel encryption model to encrypt it based on logistic mapping. This part also introduces the previous coverless information hiding algorithm proposed by [34].

2.1. Logistic Mapping

Chaos refers to the seemingly random and irregular movement that occurs in a non-deterministic system. It originates from nonlinear dynamic systems. Logistic mapping is one of the most famous chaotic mappings due to its simple expression and excellent performance [37–39]. Its expression form is as Equation (1).

$$y_{n+1} = \mu \times y_n \times (1 - y_n), \tag{1}$$

where $n = 0, 1, 2, 3, \ldots$ and y belongs to 0 and 1. The system control parameter μ is the constant and $\mu \in (0, 4]$. When $3.569945672 \cdots < \mu \leq 4$ and the final sequence value belongs to $[0, 1]$, the logistic sequence y_n is in the state of chaos. The function has the characteristics of aperiodic and sensitive dependence to the initial condition. Logistic mapping is simple without losing the complex characteristics of the chaos, so it is usually used on image encryption.

2.2. Previous Method

The main idea is to form a mapping relationship between the carrier and the secret message, and then realize the coverless hiding of the secret data. The construction of the look-up table is based on the generation of hash code. Fatimah Shamsulddin Abdulsattar generates eigenvalues based on feature decomposition, and then calculates the hash code of the image block and builds a look-up table [34]. The look-up table contains the hash code and its location information. In the image, the main feature vector represents the direction of the maximum change between image pixels, and the largest feature value is related to the foremost feature vector. The hash code is calculated by the largest eigenvalue obtained by feature decomposition which is Equation (2).

$$max\_e_{jl} = max\{e_{jl1}, e_{jl2}, e_{jl3}, \ldots, e_{jlk}\}, \tag{2}$$

where max_e_{jl} denotes the largest value in the l-th sub-block and the block $\mathbf{B_{jl}}$ has k eigenvalues. The largest eigenvalue of adjacent image sub-blocks is arranged and combined in light of the specific arrangements to obtain an 8-bit binary hash code. The arrangements in this method are displayed in Figure 1. If the arrangement Arr.1, as shown in Figure 1a, is selected in the process, the hash code is acquired by Equation (3). When another arrangement (Arr.2 or Arr.3 or Arr.4) is selected, the function is calculated according to Equation (4).

$$h_c = \begin{cases} 0, & \text{if } max\_e_{jl} > max\_e_{j5} \\ 1, & \text{if } otherwise \end{cases} \tag{3}$$

where the max_e_{jl} is the largest value in the l-th sub-block and $l \in [1,8]$.

$$h_c = \begin{cases} 0, & \text{if } max\_e_{jl} < max\_e_{jl+1} \\ 1, & \text{if } otherwise \end{cases} \tag{4}$$

where $l \in [1,8]$ and $l \neq 5$.

The hash code of each sub-block can be converted into the corresponding ASCII-code value [40]. The range of the ASCII code is $[0,255]$. The look-up table is created by putting the positions of the hash code and the hash code into the same table. When the sender hides the secret message, they can map the table with the data and record the location of the hash code, that is, by creating a location table to achieve coverless hiding. After receiving the location table and the original image, the receiver can establish the same look-up table and then extract the hidden secret from the look-up table in the light of the mapping relationship.

According to the above description, Fatimah Shamsulddin Abdulsattar uses eigenvalue decomposition to establish the hash code as the feature of the image and then establish the look-up table. The secret message is hidden according to the mapping relationship. Although their method does not require a large database and achieves a high hidden capacity, when the image block is set to a large size, the number of hash codes generated is less. There is a problem that the secret message has no mapping relationship. When the size of the secret message increases, the corresponding location table requires more space. In other words, there is room for improvement. Our method for this problem is increasing the diversity of hash code and further improving the hiding rate.

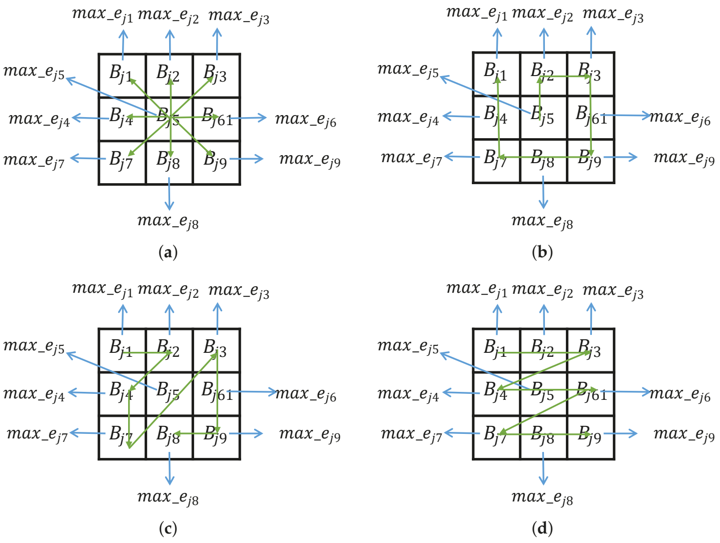


Figure 1. The four arrangements are used in [34]. (**a**) Arr.1, (**b**) Arr.2, (**c**) Arr.3, and (**d**) Arr.4.

3. Proposed Method

The information hiding technology proposed in the paper is mainly divided into two, secret message hiding and secret message extraction. The whole process is divided into two stages. The first stage takes advantage of the coverless information hiding to hide the secret message. The stage generates a vital location table. The second stage embeds the location table using the PEE algorithm. The main idea of our model is to combine coverless information hiding and PEE during hiding the secret message.

Before the secret message is hidden, the new encryption model is designed to enhance the security of secret data first. To increase the diversity of hash codes, the original image is divided into two sub-images using the two-level mechanism, and the mapping relationship formed by the look-up table established by the image blocks of the sub-image and the secret message is formed to realize the coverless hiding of the secret data. Although the hiding process does not change the original image, it needs to generate a corresponding location table. Considering that the location table needs additional storage and transmission, the paper adopts PEE to realize the second embedding of the location table. After receiving the watermarked image, the receiver can extract the location table and recover the original image. After obtaining the original image, the same look-up table can be established, and the encrypted secret information can be extracted according to the mapping relationship. The secret message can be recovered through decryption model with the same key. Figure 2 shows the framework model of the proposed algorithm. Due to the parameters used in the sender and receiver are the same, the designed model is symmetric.

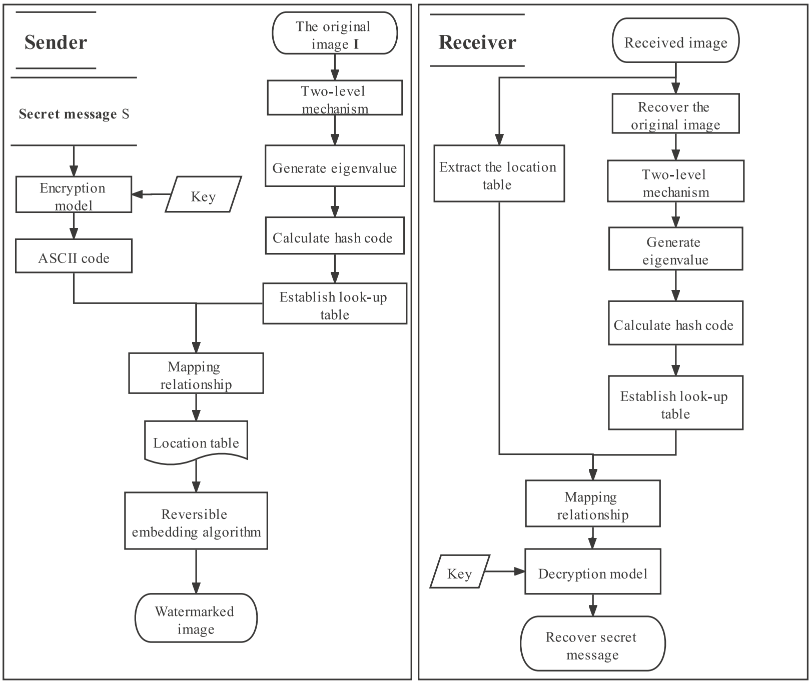


Figure 2. The framework of the proposed model.

3.1. Encrypt the Secret Message

Logistic mapping is an encryption algorithm with a simple structure and good encryption effect. However, it is sensitive and periodic to the initial value, and its structure is relatively simple. Once the initial value is known, it is easy to be cracked. Therefore, we adopt a dual logistic mapping, and the function of the model can be described as Equation (5).

$$\begin{cases} y_{i+1} = \mu \times y_i \times (1 - y_i),\ i = 1, 2, 3, \ldots, n/2 - 1 \\ y_{i+1} = \mu \times sin(\pi/2 \times y_i) \times (1 - sin(\pi/2 \times y_i)),\ i = n/2, n/2 + 1, \ldots, n - 1 \end{cases} \tag{5}$$

By combining the original logistic mapping and sin-logistic mapping, the final sequence is more chaotic. It is less easy to traverse to the secret message after encryption. If the initial value is not set, ten bits of secret information will be randomly selected to calculate the mean value as the initial value, see as Equation (6).

$$\begin{cases} r = randperm(sum, 10), \\ y_1 = \dfrac{\sum S(r)}{10}, \end{cases} \tag{6}$$

where S is the one-dimensional secret message sequence and the length is sum.

3.2. Hiding the Secret Message

The two-level mechanism splits the image into two sub-images, and two sub-images can be restored to the original image. Each pixel value of the image is composed of an 8-bit binary value, and the image can be divided into 8-bit planes. The image is split into two images according to the 8-bit planes of the image. The pixel value of the image is represented as Equation (7).

$$p = p_h \times 2^w + p_l, \tag{7}$$

where p represents the pixel value in the image, p_h represents the high-level plane, w represents the number of bits divided, and p_l is the low-level plane. According to Equation (8), the high-level plane is divided into two parts p_{h1} and p_{h2}, where a is the constant ranging from 0 to 1.

$$\begin{cases} p_{h1} = a \times p_h, \\ p_{h2} = p_h - p_{h1} \end{cases} \tag{8}$$

Equation (9) is the method of dividing the low-level plane, that is, dividing the low-level plane into two sub-planes p_{l1} and p_{l2}. The parameter c is the number of bits during the division.

$$\begin{cases} p_{l1} = \dfrac{(p_l - p_{l1})}{2^c}, \\ p_{l2} = p_l \% 2^c \end{cases} \tag{9}$$

The two sub-planes of the high plane are combined with the two sub-planes of the low plane, respectively, according to Equation (10). Thus, two sub-images can be obtained by splitting one image according to the above-mentioned division mechanism.

$$\begin{cases} p_1 = p_{h1} \times 2^w + p_{l1} \times 2^c, \\ p_2 = p_{h2} \times 2^w + p_{l2} \end{cases} \tag{10}$$

The hiding stage is mainly divided into two parts; one is coverless information hiding based on the look-up table approach, and the other embeds the location table into the original image by PEE. Firstly, the image is split into two sub-images and divided into image blocks. Secondly, the 8-bits hash code of the image block is calculated according to the designed arrangements, and each group of hash code is converted into ASCII code. The ASCII code of all image blocks and their location are combined to form a look-up table. Then the secret message is hidden without changing depending on the mapping relationship. Finally, the key location table is embedded into the image by PEE. The specific embedding steps are as follows:

Step 1. Predict the pixel value $p_{r,i}$ by Equation (11) according to the predictor as shown in Figure 3.

$$\hat{p}_{r,i} = \frac{p_{r+1,i-1} + p_{r+1,i} + p_{r+1,i+1}}{3}, \tag{11}$$

where $\hat{p}_{r,i}$ is the predicted value.

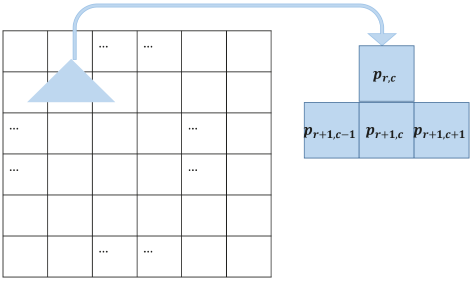


Figure 3. The predictor used in our algorithm.

Step 2. Calculate the prediction error of the pixel $p_{r,i}$ according to Equation (12).

$$e_{r,i} = p_{r,i} - \hat{p}_{r,i} \tag{12}$$

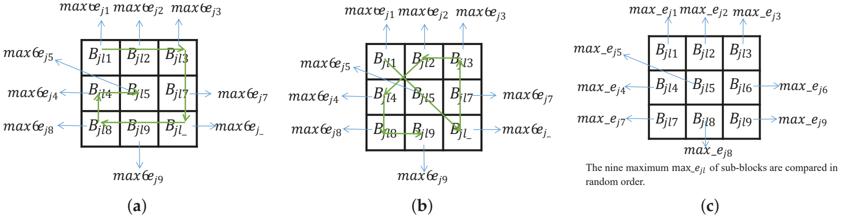
Step 3. Expand the prediction error and embed the secret message by Equation (13) where T is a threshold. The prediction error in $[-T, T)$ will be used to embed the secret message.

$$e'_{r,i} = \begin{cases} 2 \times e_{r,i} + s_i & \text{if } e_{r,i}\, \epsilon\, [-T, T) \\ e_{r,i} + T & \text{if } e_{r,i}\, \epsilon\, [T, \infty) \\ e_{r,i} - T & \text{if } e_{r,i}\, \epsilon\, (-\infty, -T) \end{cases} \tag{13}$$

Step 4. Modify the corresponding pixel value according to the extended prediction error referring to Equation (14).

$$p'_{r,i} = \hat{p}_{r,i} + e'_{r,i} \tag{14}$$

After the secret message is embedded into the original and is transmitted to the receiver, the receiver could extract the message and recover the original image by the inverse process. Our method designs three novel arrangements as seen in Figure 4 to increase the diversity of hash codes. In Figure 4c, the nine values of the sub-block are scrambled randomly to generate the hash code so that each hiding process will have different results. The paper adopts the two-level mechanism and selects three arrangements from seven ways in Figures 1 and 4 randomly to generate features. The random approach adopted can elevate the security of the architecture. This mode increases the diversity of hash code and can further allow the secret message to be successfully hidden. To understand the whole hiding process, the specific steps can be seen as Algorithm 1.


$max6e_{j1}$ $max6e_{j2}$ $max6e_{j3}$
$max6e_{j5}$
$max6e_{j4}$
$max6e_{j8}$
$max6e_{j7}$
$max6e_{j\_}$
$max6e_{j9}$
B_{jl1} B_{jl2} B_{jl3}
B_{jl4} B_{jl5} B_{jl7}
B_{jl8} B_{jl9} $B_{jl\_}$
(a)
$max6e_{j1}$ $max6e_{j2}$ $max6e_{j3}$
$max6e_{j5}$
$max6e_{j4}$
$max6e_{j8}$
$max6e_{j7}$
$max6e_{j\_}$
$max6e_{j9}$
B_{jl1} B_{jl2} B_{jl3}
B_{jl4} B_{jl5} B_{jl7}
B_{jl8} B_{jl9} $B_{jl\_}$
(b)
max_e_{j1} max_e_{j2} max_e_{j3}
max_e_{j5}
max_e_{j4}
max_e_{j7}
max_e_{j6}
max_e_{j9}
max_e_{j8}
B_{jl1} B_{jl2} B_{jl3}
B_{jl4} B_{jl5} B_{jl6}
B_{jl7} B_{jl8} B_{jl9}
(c)

The nine maximum max_e_{jl} of sub-blocks are compared in random order.

Figure 4. The remaining three arrangements we designed. (**a**) Arr.5 (**b**) Arr.6, (**c**) Arr.7.

Algorithm 1 Pseudo-code of the hiding model.

```
Require: Original image $\mathbf{I}$, secret message $S$, the initial key $Init_k$, parameter $\mu$, $w$, $c$, $b$, $b_s$
Ensure: Watermarked image $\mathbf{I}^{'}$
  Step 1. Generate one-dimensional sequence $Y$ that the length is as same as the secret message using Equations (5) and (6) with the initial key $Init_k$.
  Step 2. Encrypt the secret message $S$ by exclusive-or operation with Y and obtain the encrypted sequence $S^{'}$.
  Step 3. Divide host image $\mathbf{I}$ into two sub-images $\mathbf{I_1}$ and $\mathbf{I_2}$ based on the two-level mechanism by Equations (8)–(10).
  Step 4. Split two sub-images into image block $\mathbf{B_j}$ with fixed-size $b$.
  Step 5. Further split $\mathbf{B_j}$ into nine sub-blocks $\mathbf{B_{jl}}$ with the small size $b_s$ where $l$ denotes the $l$-th sub-block in the image block $\mathbf{B_j}$.
  Step 6. Calculate the eigenvalue of all sub-blocks $\mathbf{B_{jl}}$ and find the largest value $max\_e_{jl}$ by Equation (2).
  Step 7. Acquire the hash code by arranging the eigenvalues of the sub-blocks in every image block where uses three arrangements of seven arrangements randomly.
  Step 8. Generate the ASCII code with every 8-bit hash code and an ASCII code corresponding to a block.
  Step 9. Establish look-up table including ASCII codes and their location.
  Step 10. Convert the encrypted message $S^{'}$ to ASCII code $S_a^{'}$.
  Step 11. Match $S_a^{'}$ with the equal image block and record the location of the corresponding image block in order in the position table.
  Step 12. Embed the location table and additional information into the original image $\mathbf{I}$ using above PEE. The additional information includes the encryption key, parameter $w$, $c$, $b$, $b_s$, the arrangements selected, and the size of the secret message.
```

3.3. Extract the Secret Message and Recover the Original Image

The symmetric framework of requiring secret data is the inverse process of embedding. After receiving the watermarked image, the receiver first performs the inverse operation of the prediction error expansion to extract the location table and additional information. Then, the original image is recovered, and the same look-up table is established. Finally, the key and encrypted secret data are extracted according to the mapping relationship. The secret can be recovered by decrypting the encrypted message with the key. The pseudo-code is displayed in Algorithm 2 which includes the whole process of extracting the secret message and recovering the original image.

Algorithm 2 Pseudo-code of the extraction and recover model.

```
Require: Watermarked image
Ensure: Recovered host image $\mathbf{I_o}$ and the secret message $S^{''}$
  Step 1. Extract the location table, additional information, and recover the original image $\mathbf{I_o}$.
  Step 2. Divide the recovered image using the same method.
  Step 3. Calculate the corresponding eigenvalue and hash code.
  Step 4. Convert the hash codes to ASCII code.
  Step 5. Establish the look-up table.
  Step 6. Acquire the encrypted secret sequence from the look-up table and the location table by the mapping relation.
  Step 7. Generate one-dimensional sequence $Y_2$ that the length is as same as the secret message using Equations (5) and (6) with the initial key $Init_k$.
  Step 8. Decrypt the secret message $S^{''}$ by executing exclusive or operation between $Y_2$ and the encrypted sequence.
```

4. Experiment and Results

The computer configuration used in this article is: Intel(R) Core(TM) i5-8500 CPU @ 3.00 GHz, 16.0 GB memory, and Windows 10 (64 bits), and the experimental codes are all running on the MATLAB R2018a.

To prove the effectiveness of the encryption method, we take an example for the encryption test and compare with other encryption methods. Both methods use the same initial value to iteratively generate the sequence. Information entropy is one of the crucial indicators to measure the performance of encryption models. It is the average of the information and is expressed as Equation (15).

$$\begin{cases} I_H(p) = -\sum_{i=1}^{ps} q(p_i) \times log_2 q(p_i), \\ \sum_{i=1}^{n} q(p_i) = 1, \end{cases} \tag{15}$$

where $q(p_i)$ satisfies $0 \leq q(p_i) \leqslant 1$.

For the digital image, the information entropy can reflect the distribution of gray values. If the gray pixel value distribution is uniform, the maximum value of information entropy will be 8, which is a proportional relationship. When the information entropy is larger, the more average the gray value distribution is, the smaller the correlation degree is.

The experimental results are shown in Table 1. It can be concluded from the data that the information entropy of the original image is 7.2081, and the information entropy after encryption can exceed 7.99; more than 7.99, in theory, means that encryption is considered successful.

Table 1. Information entropy and correlation coefficient compared with other encryption model.

Index\Image	**Original Image**	**Encrypted Image**		
		Logistic Mapping [38]	**Sin-Logistic Mapping**	**Our Model**
Entropy	7.2081	7.9945	7.9623	**7.9954**
Horizontal	0.9687	−0.0868	0.0620	**0.0352**
Vertical	0.9372	−0.0934	−0.1215	**−0.0391**
Diagonal	0.9057	0.0722	0.0514	**0.0188**

In Table 1, the information entropy of the encrypted image is 7.9954, which is closer to 8 than the other two algorithms, which proves that the method can have a more average gray value distribution after application. Our algorithm achieves the maximum information entropy, which means it performs excellently compared with other methods.

Each digital image is not independent, and the correlation between adjacent pixels is crucial. The calculation function is as Equation (16). One purpose of the image encryption is reducing the correlation between pixels, making correlation analysis invalid. The smaller the correlation value between pixels, the better the encryption effect and the safer the information. A strong correlation must be broken to avoid the statistical attack. So, we design correlation analysis experiments in three different directions and the directions include horizontal, vertical, and diagonal direction. The data are in Table 1.

$$\begin{cases} C_R(p, p_x) & = \frac{cov(p,p_x)}{\sigma_p \times \sigma_{p_x}}, \\ cov(p, p_x) = & \frac{\sum_{j=1}^{M}(p_j - E(p)) \times (p_{x_j} - E(p_x))}{M} \\ \sigma_p = & \frac{\sum_{j=1}^{M}(p_j - E(p))^2}{M}, \\ E(p) = & \frac{\sum_{j=1}^{M} p_j}{M} \end{cases} \tag{16}$$

So, we design correlation analysis experiments in three different directions where 2000 pairs of adjacent pixels are randomly selected for testing. The correlation coefficients in the original image and the encrypted image are calculated in the horizontal, vertical, and diagonal directions according to the above definition of the correlation coefficient. The data are in Table 1. According to Equation (16), it can be known that the coefficient may be positive or negative. When $|C_R| \leq 0.3$, it means that the correlation between the two variables is extremely weak and can be regarded as irrelevant. From Table 1, we can see that the correlation coefficient of the original image is close to 1, which means that there is a high correlation between pixels. On the contrary, the correlation coefficient of the encrypted image is close to 0, which means that the statistical characteristics of the encrypted image are successfully disrupted. As shown, our method achieves better dispersion than other algorithms in the horizontal, vertical, and diagonal direction and holds higher security.

Information hiding should realize the complete hiding of the secret message under the premise of ensuring the quality of the original image. In [34], the experimental results prove that there will be cases where secret message cannot be found to match it in the look-up table. Hash codes are important features used in coverless information hiding. When the range of generated features is large and different enough, hash codes are more diverse. The paper designs new arrangements, and there are seven arrangements now. In addition, we employ the two-level mechanism. Comparative experiments are performed on three different images, as shown in Figure 5. In the same figure, the experiment compares the generated types of hash codes by seven arrangements.

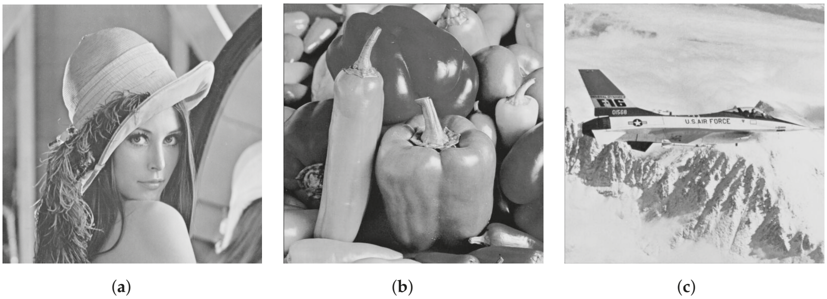


Figure 5. The three test images. (**a**) Lena, (**b**) Pepper, (**c**) Plane.

The results are put in Table 2 which has only one arrangement. From the data in Table 2, we see that the types of hash codes generated under the two-level mechanism become larger, that is, the designed model can find more different hash codes. The diversity of the hash code is enhanced. However, the types of hash codes generated under a single arrangement still cannot reach 256. Therefore, the paper increases the formation range of hash codes and increases the diversity of hash code through the two-level mechanism and three new arrangements to achieve 256 different hash codes. The method can generate enough unique features in a non-overlapping way. To prove the effectiveness, a comparative experiment is carried out on Figure 5 when hiding the same secret message.

From the data, we can see that the hash code value generated under the two-level mechanism is larger, that is, the designed model can find more different hash codes. The diversity of the hash code is enhanced. However, the types of hash codes generated when a single arrangement is adopted still cannot reach 256. Therefore, the paper increases the formation range of hash codes and increases the diversity of hash code through the two-level mechanism and three new arrangements to achieve 256 different hash codes. The method can generate enough unique features in a non-overlapping way. To prove the effectiveness, a comparative experiment is carried out on Figure 5 when hiding the same secret message. The size of the secret message is set as 6272 bits. The test and comparison experiment results are shown in Table 3.

Table 2. Types of different hash codes generated under different arrangements.

Hash Codes	[34]			Two-Level Mechanism		
	Lena	Pepper	Plane	Lena	Pepper	Plane
Arr.1	148	139	158	186	161	198
Arr.2	179	180	214	186	182	207
Arr.3	155	163	169	170	161	186
Arr.4	208	188	158	251	166	248
Arr.5	192	191	201	217	210	227
Arr.6	171	160	195	170	166	191
Arr.7	143	123	170	183	176	210

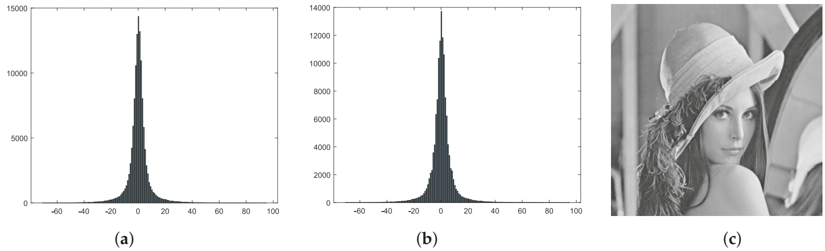
In the experiment, the size of the secret message is set as 6272 bits. The test and comparison experiment results are shown in Table 3. Arr.4 is used in [34] and our method selected three arrangements including Arr.2, Arr.4, and Arr.5. In Table 3, "Hash code(types)" denotes the types of hash code generated, and "No-find(bits)" is that the secret message cannot find the corresponding hash code. Since every 8 bits of secret information are converted into ASCII code for embedding, the ASCII-code range is between 0 and 255. Therefore, the hash code generated should be converted into ASCII code in 256 cases. It can be seen from the test data that the previous method cannot produce enough types of hash code. There is some message that cannot find the mapping relationship in the look-up table. Our algorithm obtains more different hash codes, that is, the diversity of hash code is increased by adding arrangements and the two-level mechanism. Due to the diversity of hash code increasing, the number of no-find cases will decrease accordingly. We increase the concealment rate of secret message through adding the two-level mechanism and designing new arrangements. As the results in Table 3, when the same number of bits is hidden in the same image, we can hide all secret message compared with the previous method.

Table 3. Comparison of the types of hash code and the number of the secret message cannot find.

Hiding Capability	[34]		Our Method	
	Hash Code (Types)	No-Find (Bits)	Hash Code (Types)	No-Find (Bits)
Lena	208	156	**256**	**0**
Pepper	188	219	**256**	**0**
Plane	158	305	**256**	**0**

Our coverless information hiding method can store an infinite amount of secret information, but for ensuring the security and transmission space of the location table, we also use reversible information hiding technology to embed the location table into the original image. Therefore, the embedding capacity depends on the method of reversible information hiding technology used. This paper adopts a simple PPE algorithm. After hiding the secret data, the image carrying the secret message is put in Figure 6. The secret message cannot be detected from Figure 6c by our eyes.

Reference [34] realizes the high capacity. The high hiding capacity is one of our goals in the paper. The experiment of the capacity in the paper is compared with other methods in Figure 5a, "Lena". The test results are stored in Table 4, where the different image has the different capacity with different embedding algorithms. The capacity represents the largest length of secret message that can be hidden under the algorithm model. From the data in Table 4, our method has the highest capacity to hide the secret message. Regarding reversible information hiding technology, it can also be combined with technologies with higher hiding ability. In the future, different reversible information hiding technologies can be selected according to the size of secret data.



(**a**) (**b**) (**c**)

Figure 6. The image with the secret message applying our method. (**a**) The histogram of the predict error; (**b**) the histogram of predict error after embedding message; (**c**) the image carrying the secret message.

Table 4. Comparison of the hiding capacity with other papers.

Approaches	Capacity (bits)
HOGs [41]	8
DCT+LDA [42]	1–15
faster-RCNN [43]	20 and 25
[34] (non-overlapping)	6272
[34] (overlapping)	55,112
Our proposed method	**84,005**

In view of analyzing the image quality, this experiment is compared with several methods. This paper still uses the standard Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) to measure the quality of several methods. PSNR is one of the criteria for measuring the invisibility of images embedded with watermarks in information hiding, calculated by Equation (17).

$$\begin{cases} \text{PSNR} = 10 \times log_{10}(\dfrac{\text{MAX}_I^2}{\text{ME}}), \\ \text{MAX}_I = 2^8 - 1, \\ \text{ME} = \dfrac{\sum_{u=1}^{rol}\sum_{v=1}^{col}(p_{uv} - p_{uv}^{'})^2}{rol \times col}, \end{cases} \tag{17}$$

where p_{uv} and $p_{uv}^{'}$, respectively, are the pixel value in the original host and the stego image. *rol* and *col* denote the total number of image rows and columns.

SSIM is an index to judge the similarity between two images, and the calculation function can be seen in Equation (18). Here, SSIM is used to measure the extraction quality of secret information.

$$\text{SSIM}(\mathbf{I}, \mathbf{I}^{'}) = \frac{(2 \times \mu_I \times \mu_{I'} + c_{a1})(2 \times \sigma_{II'} + c_{a2})}{(\mu_I^2 + \mu_{I'}^2 + c_{a1})(\sigma_I^2 + \sigma_{I'}^2 + c_{a2})}, \tag{18}$$

where μ_I and $\mu_{I'}$ represent the average of I and $I^{'}$. $\sigma_{I'}$ and $\sigma_{I'}^2$ is the variance of $\mathbf{I}$ and $\mathbf{I}^{'}$. $\sigma_{II'}$ is the covariance. c_{a1} and c_{a2} are two constants to avoid dividing by zero.

Table 5 shows PSNR and SSIM for different methods without suffering any attacks. From the calculation results, we can see that our method is the best performance of these methods. This algorithm achieves excellent image quality when it has the same capacity and can obtain the original secret message.

Table 5. PSNR and SSIM about our proposed method and other papers.

Methods	PSNR	SSIM
Sahu and Swain [5]	51.25	0.999
Muhuri et al. [6]	51.668	0.998
Sahu and Swain [43]	48.2	0.997
[34]	∞	**1**
Our method	52.024	**1**

In the network transmission, images may be intercepted or tampered with, so the information hiding algorithm needs to have the ability to resist attacks, which we call robustness. The robustness refers to the nature of an image with the secret message that can still extract information after suffering the attack. To reflect the robustness of the model designed, we compare it with the previous model. The experimental results are shown in Table 6.

Table 6. The extraction rate using different methods under different attacks.

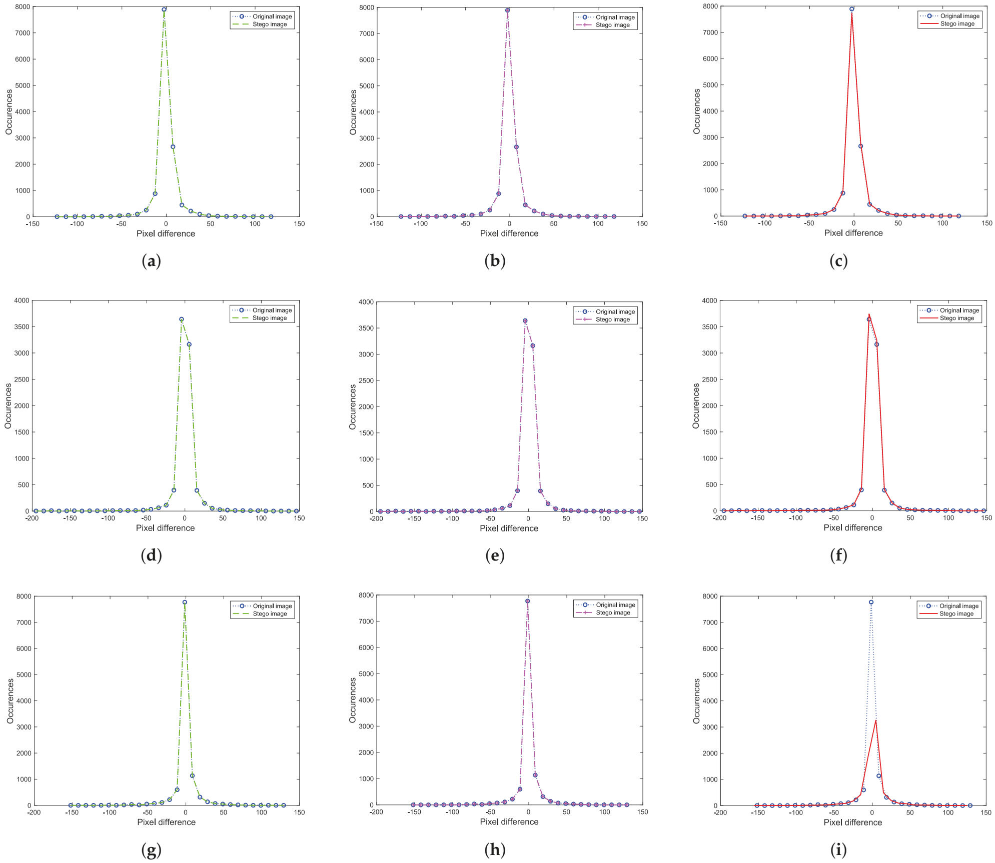
Attacks	[34]	Our Hiding Method
No attack	80.10%	**100%**
Median (3 × 3)	65.90%	**72.60%**
Median (5 × 5)	59.60%	**68.20%**
Gaussian low-pass filter (w = 3)	69%	**75.80%**
Gaussian noise (r = 0.001)	**39.30%**	6.90%
Salt and pepper noise	**3.70%**	1%
Speckle noise	**39.90%**	10.60%
Sharpening attack (r = 0.05)	**88.50%**	78.80%
Histogram equalization	0.80%	0.80%
Average filter	69%	**77.30%**
Motion blur	58.50%	**59.70%**

It can be seen from the extraction rate that under the same hidden capacity, our algorithm has higher value, which means that our model has better robustness.

Pixel Difference Histogram (PDH) is an important indicator to measure security [44,45]. The PDH graph reveals the relationship between the pixel difference and the number of occurrences of the difference. The X-axis represents the pixel difference between two consecutive pixels, and the Y-axis is the frequency of the difference. The zig-zag phenomenon that appears in the PDH curve is called the undesired step effect. If this effect appears in the curve, it is considered that the image hides the secret message. The greater the distortion of the image containing the secret message, the corresponding PDH curve shows undesired steps; conversely, when the distortion is very small, the PDH curve would look as smooth as the original image.

The same secret message is hidden in three test images located in Figure 5. Figure 7a,d,g are the curves using [34]. In this experiment, the first stage (coverless information hiding) and the final image that carries the location table of our algorithm are represented in the analysis, respectively. Figure 7b,e,f are the first stage and Figure 7c,f,i denote the results in the final image.

It can be seen that these PDH curves are still as smooth as the original image after hiding the secret message and they do not show the effects of any steps. Therefore, this algorithm is undetectable by PDH analysis and is considered to have a certain degree of security.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 7. The PDH analysis with different images: (**a**–**c**) are the results in "Lena", (**d**–**f**) are the results in "Pepper", and (**g**–**i**) are the results in "Plane".

5. Conclusions

Network transmission is a vital part of our modern life. With the convenience it brings, information security issues follow. Information hiding has made great progress. Considering the problems of the previous work requires a large number of training databases, has the low hiding rate, and occupies a big space. This work adopts a two-level mechanism, look-up table, and reversible information hiding technology based on high-quality information hiding. Through reversible information hiding, the additional storage and transmission burden is solved. In addition, to further improve the security of the secret message, a new encryption model is designed. Compared with different methods in the experiment, our proposed model achieves high capacity and perfect hiding rate under the premise of ensuring image quality. The robustness is achieved in the attack tests. To verify the security, the PDH analysis is also performed. The test results prove that the method is undetectable by PDH. Therefore, our algorithm can achieve better results in terms of image quality, capacity, and security compared with other methods. The proposed symmetry model is outstanding and efficient.

Furthermore, we can combine different methods with higher capacity and better robustness in future work. In addition, more diverse hash codes or other feature values can be designed to hide the secret message. The method is efficient in the paper, but once an attacker masters our model, hidden information can be extracted or even tampered with.

So, to further improve the security of secret messages, asymmetric mechanisms are also the focus of our future research.

Author Contributions: Conceptualization, X.-X.S. and J.-S.P.; formal analysis, J.-S.P., X.-X.S., V.S. and S.-C.C.; methodology, J.-S.P., X.-X.S., S.-C.C. and H.Y.; validation, J.-S.P., H.Y. and V.S.; writing—original draft preparation, X.-X.S.; writing—review and editing, J.-S.P., X.-X.S., S.-C.C., V.S. and H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

I	Original image
I$'$	Watermarked image
S	Secret message
$Init_k$	Initial key on the encryption algorithm
Y	One-dimensional sequence used for encryption
y_n	n-th value in Y
μ	System control parameter
e_{jlk}	The k-th eigenvalue in the l-th sub-block
max_e_{jl}	The largest value in all eigenvalues of the l-th sub-block
$\mathbf{B}_{jl}$	The l-th sub-block of j-th block
h_c	Hash code
sum	The length of secret message
p	The pixel value in the image
p_h	The high-level plane
p_l	The low-level plane
w	The number of bits divided in the plane
$\hat{p}_{r,i}$	The predicted pixel value
$p_{r,i}$	The pixel value of the original image at row r and column i
$e_{r,i}$	The error value between the pixel value and the predicted value
T	Threshold
$e'_{r,i}$	The modified error value
$p'_{r,i}$	The modified pixel value
s_i	One bit in secret message S
$I_H(p)$	The average of the information
$C_R(p, p_x)$	Correlation coefficient between p and p_x
p_{uv}	The pixel value in the original image
p'_{uv}	The pixel value in the stego image
rol	The total rows of image
col	the total columns of image
PSO	Particle Swarm Optimization
IWT	Integer Wavelet Transformation
PEE	Prediction Error Expansion
PDH	Pixel Difference Histogram

References

1. Pfleeger, C.P. The fundamentals of information security. *IEEE Softw.* **1997**, *14*, 15–16. [CrossRef]
2. Shelupanov, A.; Evsyutin, O.; Konev, A.; Kostyuchenko, E.; Kruchinin, D.; Nikiforov, D. Information Security Methods—Modern Research Directions. *Symmetry* **2019**, *11*, 150. [CrossRef]
3. Mei, X.S.; Chen, H.T.; Fan, H.Y.; Lu, Z.M.; Yeh, J.h. A Robust Digital Image Watermarking Scheme for Content Protection. *J. Netw. Intell.* **2020**, *5*, 54–61.
4. Zhang, Z.; Chen, S.; Sun, X.; Liang, Y. Trajectory privacy protection based on spatial-time constraints in mobile social networks. *J. Netw. Intell.* **2021**, *6*, 485–499.
5. Sahu, A.K.; Swain, G. A novel n-rightmost bit replacement image steganography technique. *3D Res.* **2019**, *10*, 2. [CrossRef]

6. Muhuri, P.K.; Ashraf, Z.; Goel, S. A novel image steganographic method based on integer wavelet transformation and particle swarm optimization. *Appl. Soft Comput.* **2020**, *92*, 106257. [CrossRef]
7. Song, P.C.; Chu, S.C.; Pan, J.S.; Yang, H. Simplified Phasmatodea population evolution algorithm for optimization. *Complex Intell. Syst.* **2021**, 1–19. [CrossRef]
8. Pan, J.S.; Song, P.C.; Pan, C.A.; Abraham, A. The Phasmatodea Population Evolution Algorithm and Its Application in 5G Heterogeneous Network Downlink Power Allocation Problem. *J. Internet Technol.* **2021**, *22*, 1199–1213.
9. Pan, J.S.; Sun, X.X.; Chu, S.C.; Abraham, A.; Yan, B. Digital watermarking with improved SMS applied for QR code. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104049. [CrossRef]
10. Ni, Z.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–362.
11. Luo, H.; Chu, S.C.; Lu, Z.M. Self embedding watermarking using halftoning technique. *Circuits Syst. Signal Process.* **2008**, *27*, 155–170. [CrossRef]
12. Shi, Y.Q.; Li, X.; Zhang, X.; Wu, H.T.; Ma, B. Reversible data hiding: Advances in the past two decades. *IEEE Access* **2016**, *4*, 3210–3237. [CrossRef]
13. Nguyen, T.D.; Le, H.D. A reversible data hiding scheme based on (5, 3) Hamming code using extra information on overlapped pixel blocks of grayscale images. *Multimed. Tools Appl.* **2021**, *80*, 13099–13120. [CrossRef]
14. Linb, C.Y.Y.C.H.; Hua, W.C. Reversible data hiding for high-quality images based on integer wavelet transform. *J. Inf. Hiding Multimed. Signal Process.* **2012**, *3*, 142–150.
15. Honsinger, C.W.; Jones, P.W.; Rabbani, M.; Stoffel, J.C. Lossless Recovery of an Original Image Containing Embedded Data. U.S. Patent 6,278,791, 21 August 2001.
16. Hu, Y.; Lee, H.K.; Li, J. DE-based reversible data hiding with improved overflow location map. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *19*, 250–260.
17. Faragallah, O.S.; Elaskily, M.A.; Alenezi, A.F.; El-sayed, H.S.; Kelash, H.M. Quadruple histogram shifting-based reversible information hiding approach for digital images. *Multimed. Tools Appl.* **2021**, *80*, 26297–26317. [CrossRef]
18. Weng, S.; Zhao, Y.; Pan, J.S.; Ni, R. A novel reversible watermarking based on an integer transform. In Proceedings of the 2007 IEEE International Conference on Image Processing, San Antonio, TX, USA, 16 September–19 October 2007; Volume 3, p. III-241.
19. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. [CrossRef]
20. Alattar, A.M. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156. [CrossRef] [PubMed]
21. Thodi, D.M.; Rodriguez, J.J. Prediction-error based reversible watermarking. In Proceedings of the 2004 International Conference on Image Processing, ICIP'04, Singapore, 24–27 October 2004; Volume 3, pp. 1549–1552.
22. Tseng, H.W.; Hsieh, C.P. Prediction-based reversible data hiding. *Inf. Sci.* **2009**, *179*, 2460–2469. [CrossRef]
23. Weng, S.; Chu, S.C.; Cai, N.; Zhan, R. Invariability of Mean Value Based Reversible Watermarking. *J. Inf. Hiding Multim. Signal Process.* **2013**, *4*, 90–98.
24. Weng, S.; Zhao, Y.; Ni, R.; Pan, J.S. Lossless data hiding based on prediction-error adjustment. *Sci. China Ser. F Inf. Sci.* **2009**, *52*, 269–275. [CrossRef]
25. Yu, Y.; Lei, M.; Liu, X.; Qu, Z.; Wang, C. Novel zero-watermarking scheme based on DWT-DCT. *China Commun.* **2016**, *13*, 122–126. [CrossRef]
26. Weng, S.; Chen, Y.; Hong, W.; Pan, J.S.; Chang, C.C.; Liu, Y. An improved integer transform combining with an irregular block partition. *Symmetry* **2019**, *11*, 49. [CrossRef]
27. Yan, Y.S.; Cai, H.L.; Yan, B. Data Hiding in Symmetric Circular String Art. *Symmetry* **2020**, *12*, 1227. [CrossRef]
28. Pan, J.S.; Luo, H.; Lu, Z.M. Look-up table based reversible data hiding for error diffused halftone images. *Informatica* **2007**, *18*, 615–628. [CrossRef]
29. Zhou, Z.; Su, Y.; Zhang, Y.; Xia, Z.; Du, S.; Gupta, B.B.; Qi, L. Coverless Information Hiding Based on Probability Graph Learning for Secure Communication in IoT Environment. *IEEE Internet Things J.* **2021**. [CrossRef]
30. Zhou, Z.; Cao, Y.; Wang, M.; Fan, E.; Wu, Q.J. Faster-RCNN Based Robust Coverless Information Hiding System in Cloud Environment. *IEEE Access* **2019**, *7*, 179891. [CrossRef]
31. Zou, L.; Sun, J.; Gao, M.; Wan, W.; Gupta, B.B. A novel coverless information hiding method based on the average pixel value of the sub-images. *Multimed. Tools Appl.* **2019**, *78*, 7965–7980. [CrossRef]
32. Cao, Y.; Zhou, Z.; Sun, X.; Gao, C. Coverless information hiding based on the molecular structure images of material. *Comput. Mater. Contin.* **2018**, *54*, 197–207.
33. Wang, Y.; Wu, B. An intelligent search method of mapping relation for coverless information hiding. *J. Cyber Secur.* **2020**, *5*, 48–61.
34. Abdulsattar, F.S. Towards a high capacity coverless information hiding approach. *Multimed. Tools Appl.* **2021**, *80*, 18821–18837. [CrossRef]
35. Weng, S.; Pan, J.s.; Li, L. Reversible data hiding based on an adaptive pixel-embedding strategy and two-layer embedding. *Inf. Sci.* **2016**, *369*, 144–159. [CrossRef]
36. Wu, T.Y.; Chen, C.M.; Wang, K.H.; Pan, J.S.; Zheng, W.; Chu, S.C.; Roddick, J.F. Security Analysis of Rhee et al.'s Public Encryption with Keyword Search Schemes: A Review. *J. Netw. Intell.* **2018**, *3*, 16–25.
37. Andrecut, M. Logistic map as a random number generator. *Int. J. Mod. Phys. B* **1998**, *12*, 921–930. [CrossRef]

38. Pareek, N.K.; Patidar, V.; Sud, K.K. Image encryption using chaotic logistic map. *Image Vis. Comput.* **2006**, *24*, 926–934. [CrossRef]
39. Li, C.; Xie, T.; Liu, Q.; Cheng, G. Cryptanalyzing image encryption using chaotic logistic map. *Nonlinear Dyn.* **2014**, *78*, 1545–1551.
40. Elshoush, H.T.; Ali, I.A.; Mahmoud, M.M.; Altigani, A. A novel approach to information hiding technique using ASCII mapping based image steganography. *J. Inf. Hiding Multimed. Signal Process.* **2021**, *12*, 65–82.
41. Zhou, Z.; Wu, Q.J.; Yang, C.N.; Sun, X.; Pan, Z. Coverless image steganography using histograms of oriented gradients-based hashing algorithm. *J. Internet Technol.* **2017**, *18*, 1177–1184.
42. Zhang, X.; Peng, F.; Long, M. Robust coverless image steganography based on DCT and LDA topic classification. *IEEE Trans. Multimed.* **2018**, *20*, 3223–3238. [CrossRef]
43. Sahu, A.K.; Swain, G. Reversible image steganography using dual-layer LSB matching. *Sens. Imaging* **2020**, *21*, 1–21. [CrossRef]
44. Pradhan, A.; Sahu, A.K.; Swain, G.; Sekhar, K.R. Performance evaluation parameters of image steganography techniques. In Proceedings of the 2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS), Bangalore, India, 6–7 May 2016; pp. 1–8.
45. Swain, G. Very high capacity image steganography technique using quotient value differencing and LSB substitution. *Arab. J. Sci. Eng.* **2019**, *44*, 2995–3004. [CrossRef]

symmetry

MDPI

Article

A Procedure for Tracing Chain of Custody in Digital Image Forensics: A Paradigm Based on Grey Hash and Blockchain

Mohamed Ali [1], Ahmed Ismail [1], Hany Elgohary [2], Saad Darwish [3,*] and Saleh Mesbah [4]

[1] Department of Forensic Medicine, Ministry of Justice, Higher Institute for Tourism, Hotels and Computer, Al-Seyouf, Alexandria 21533, Egypt; admin@onehoster.com (M.A.); Gisapp13@gmail.com (A.I.)
[2] Expert Counterfeiting and Forgery Research, Department of Forensic Medicine, Ministry of Justice, Alexandria 21519, Egypt; hany_elgohary44@yahoo.com
[3] Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Alexandria 21526, Egypt
[4] College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria 1029, Egypt; saleh.mesbah@aast.edu
* Correspondence: saad.darwish@alexu.edu.eg; Tel.: +20-01222632369

Abstract: Digital evidence is critical in cybercrime investigations because it is used to connect individuals to illegal activity. Digital evidence is complicated, diffuse, volatile, and easily altered, and as such, it must be protected. The Chain of Custody (CoC) is a critical component of the digital evidence procedure. The aim of the CoC is to demonstrate that the evidence has not been tampered with at any point throughout the investigation. Because the uncertainty associated with digital evidence is not being assessed at the moment, it is impossible to determine the trustworthiness of CoC. As scientists, forensic examiners have a responsibility to reverse this tendency and officially confront the uncertainty inherent in any evidence upon which they base their judgments. To address these issues, this article proposes a new paradigm for ensuring the integrity of digital evidence (CoC documents). The new paradigm employs fuzzy hash within blockchain data structure to handle uncertainty introduced by error-prone tools when dealing with CoC documents. Traditional hashing techniques are designed to be sensitive to small input modifications and can only determine if the inputs are exactly the same or not. By comparing the similarity of two images, fuzzy hash functions can determine how different they are. With the symmetry idea at its core, the suggested framework effectively deals with random parameter probabilities, as shown in the development of the fuzzy hash segmentation function. We provide a case study for image forensics to illustrate the usefulness of this framework in introducing forensic preparedness to computer systems and enabling a more effective digital investigation procedure.

Keywords: blockchain; chain of custody; digital evidence; digital forensics; fuzzy hash; image forensic

Citation: Ali, M.; Ismail, A.; Elgohary, H.; Darwish, S.; Mesbah, S. A Procedure for Tracing Chain of Custody in Digital Image Forensics: A Paradigm Based on Grey Hash and Blockchain. *Symmetry* **2022**, *14*, 334. https://doi.org/10.3390/sym14020334

Academic Editors: Ming-Chin Chuang and Chin-Ling Chen

Received: 8 December 2021
Accepted: 27 January 2022
Published: 6 February 2022

1. Introduction

Any digital data containing trustworthy information that supports an event hypothesis is considered digital evidence. Digital evidence's extent is continuously increasing, including both established and emerging technology such as computers, networks, memory, and mobile devices [1]. Digital evidence has many features, including the ease with which it can be copied and transferred, the ease with which it can be changed and deleted, the ease with which it may be tainted by new data, and the fact that it is time-sensitive. Additionally, digital evidence may be easily transferred across countries. As a result, managing digital evidence is much more complex than processing physical evidence [2]. Digital evidence may take the form of images, videos, text, or device logs. Additionally, it incorporates data from social media platforms such as Twitter, Instagram, and Facebook [3–10].

There are many ways for enhancing the integrity of digital evidence. These techniques include cyclic redundancy checking, hashing functions, digital signatures, time stamps, encryption, and watermarking. Each technique has a number of benefits and drawbacks; see [8,11–14] for more details. The majority of digital forensic tools and apps use some kind of hashing algorithm to ensure the integrity of digital evidence. Hashing is a cryptographic method for determining an entity's unique representation. When utilizing the conventional hash, certain problems will occur, particularly regarding data integrity since digital data can readily altered. Tampering will always be a problem. This occurs as a result of the exchange procedure being poorly documented [15]. Additionally, a conventional hash cannot be utilized to calculate similarity or to identify traces of evidence. Fuzzy hashing is a kind of hashing that is used to determine the degree to which two entities are similar. Fuzzy hashing enables the investigator to concentrate on possibly incriminating images that would not be seen using conventional hashing techniques.

Meanwhile, a Chain of Custody (CoC) is a critical process in the management of evidence and investigations. CoC is a term that refers to the process of preserving and documenting the chronological history of digital evidence [4–6]. CoC and integrity of digital evidence play a part in the digital process of forensic investigation since forensic investigators must know where, when, and how digital evidence was found, gathered, tracked, handled, and preserved throughout its trip to a court of law. A proper CoC must include documentation that addresses each of these points. If any one of these questions is left unanswered, the CoC is compromised and disturbed. Without a certificate of conformity, the evidence is useless [7–15].

There are many indications that may be used to identify problems with the management of CoC [6,16–19]: (1) threats to the data integrity of digital evidence throughout its lifetime; (2) a massive amount of data is produced by billions of linked devices and must be stored, presenting significant difficulties in ensuring authenticity; (3) because digital evidence is complicated and volatile, and may be altered inadvertently or incorrectly after acquisition, the CoC must guarantee that the evidence gathered is admissible in court; (4) as the number of devices and types of software in the computer and information technology fields continues to increase, cybercrime faces difficulties in terms of the amount of evidence being examined; (5) documentation of the CoC is secure. This is a critical problem since digital evidence may be copied and transferred to other systems; and (6) CoC adaptability and capacity, which comes as a result of the growing amount of data produced by different new digital forensics technologies.

To address the aforementioned issues, an integrated system is required. This system must be capable of presenting data with established integrity and storing CoC for digital evidence, providing an auditing facility to ensure the accuracy of forensic tools and their application procedures, and preserving the artifacts of the evidence, in order for digital evidence to be admissible in court [6,15]. The blockchain may be used to verify the validity and legality of the processes used to collect, store, and transmit digital evidence, as well as to offer a consolidated view of all CoC interactions [20].

In its simplest form, a blockchain is a collection of linked data structures called blocks that store or monitor the state of any distributed system on a peer-to-peer network. Each block is connected to the previous block via a special pointer called a hash pointer, resulting in an append-only system, a permanent and irreversible history that can be used as a real-time audit trail by any participant to verify the accuracy of the records simply by reviewing the data itself [9]. The blockchain has been extensively utilized in a variety of areas, including cloud security, IoT security, and digital forensics. Blockchain technology is also a potential method for evidence verification and management in the area of digital forensics, and it is being extensively explored [10].

Digital image forgeries are becoming more prevalent today since image manipulation software is widely accessible and the usage of digital images has grown in popularity. One cannot tell if the image is genuine or has been altered. Images may be altered by removing a portion of the image, hiding an area within the image or altering the image in such a way that the image information is misrepresented. These flaws erode the validity of digital images [4]. Numerous methods are discussed in detail in order to identify image forgery.

They are categorized as active or passive algorithms [5]. The active method involves embedding a watermark into the picture. Because embedding watermarks in images needs specially equipped cameras, this technique is very restricted in practice. In contrast, passive methods to forgery detection rely on the evidence left on the image by various processing stages during image modification. Passive may also be used to detect the amount and location of forgeries in an image.

To summarize, computer forensics professionals use forensic software to acquire copies or images of electronic equipment and to capture associated data. Recent advances in forensic software allow for remote gathering and analysis. Even if it is impossible to precisely quantify the uncertainty inherent in a piece of digital evidence, courts should consult experts to get a sense of the data's reliability. Every piece of digital data has some degree of uncertainty, and an expert should be able to describe and estimate the degree of certainty that can be put on a particular piece of evidence. If we do not attempt to quantify uncertainty in digital evidence, one might argue that there is no foundation for assessing the evidence's dependability or correctness. Additionally, forensic examiners who do not account for ambiguity throughout their analysis risk arriving at incorrect conclusions during the investigation stage and finding it more difficult to defend their claims when cross-examined.

This paper focuses on the research of protecting digital evidence that is uncertain, which is still a challenging research topic that has not been studied much by researchers. Traditional blockchain-based chain of custody is mainly based on a concise description of the evidence under examination and some kind of hash code. However, the conventional hash method is inefficient at dealing with identical files that arise from benign or malicious alteration of the images examined by the forensic investigator. Utilizing fuzzy hash functions enables forensic investigators to successfully deal with permissible alteration to digital evidence, while using conventional hash methods is ineffective in this situation.

The remainder of this paper will be structured as follows. Section 2 discusses several similar works and their benefits and drawbacks. The suggested framework is described in Section 3. Section 4 outlines the experiments used to verify the proposed framework, and Section 5 concludes the paper.

2. Literature Review

Numerous methods have been presented to enhance the quality of CoC. Several blockchain-based secure digital evidence systems have been suggested in recent years. The authors in [21] suggested a Blockchain-based Chain of Custody (B-CoC) to dematerialize the CoC procedure while ensuring the integrity of gathered evidence and owner traceability. B-CoC was shown to effectively assist the CoC process during the performance assessment. However, the degree of anonymity for validators must be increased without modifying security attributes. In a similar manner, the authors in [15] integrated the Digital Evidence Cabinet (DEC) architecture with Blockchain. This prototype is referred as (B-DEC). B-DEC makes use of data storage integrity to handle digital evidence that relates to DEC. DEC is written in an XML format. However, the system must be capable of securely storing digital evidence through software. It needs to significantly strengthen the protection of digital evidence, such as via the use of encryption.

The work in [8] established a reliable time-stamping technique for protecting digital evidence during the investigative process. Timestamps are acquired from a secure third party in order to establish the date and time of the staff's access to the evidence. A significant issue here is that a reliable source of time is contingent on the setting of the clock that produces it. Another similar study is [12], in which the authors utilized a variety of security techniques to protect the integrity of the digital evidence, including (CRC—Hash Functions—Digital Signatures). SHA512 was chosen for integrity protection based on tests and evaluations since it is computationally extremely fast and least susceptible. However, one may alter the original data, recalculate the hash, and then exchange the original hash with the recalculated one, thus subverting the integrity service.

The authors in [19] encrypted the XML structure on the digital chain of custody data storage using the RC4 cryptography technique. One benefit of utilizing XML is that it is simple for non-professionals to comprehend. Another issue is that XML does not need a specific database management system to be opened. On the other hand, since the material is accessible to everyone, the integrity of digital evidence cannot be accepted in court. Additionally, RC4 encryption will take longer if the plaintext is lengthy. The researchers in [22] evaluated two automated disk imaging programs (Encase and FTK Imager). These programs claim that they protect the integrity of digital evidence by computing MD5 and SHA1 hashes of extracted data. The offered solution is both effective and practical. However, MD5 and SHA1 hashes are insufficient to ensure the evidence's integrity.

Z. Tian et al. [10] proposed a secure Digital Evidence Framework (Block-DEF) based on Blockchain technology, with a loose coupling structure in which evidence and evidence information are stored independently. The Blockchain is used to keep just the evidence information, which is then kept on a trustworthy storage platform. Experiments demonstrated that Block-DEF is a scalable framework; it ensures the authenticity of evidence and strikes an appropriate balance between privacy and traceability. However, when adding a new node to the blockchain it takes an inordinate amount of time to download and validate the blockchain.

While earlier blockchain-based image forensics systems employed standard hashing, the suggested approach uses fuzzy hashing to examine the blockchain validity (evidence items) in order to better handle evidence item alterations induced by both benign and malicious cyberattacks. When the similarity between two blocks surpasses 95%, the block is considered to be original.

3. Methodology

This section explains the suggested methodology for integrating digital evidence in the presence of certain defects (uncertainty of integrity) for many versions of the same document. The phase of data gathering encompasses all image forensic-capture methods. To maintain CoC throughout this phase, the examiner must adhere to forensic standards while acquiring data sources (e.g., hard drives, network packet captures, OS and application logs, memory contents, and mobile devices). With respect to the CoC, blockchain technology, especially when combined with fuzzy hashing, has the potential to provide tamper-proof recording of evidence. By using fuzzy hash functions, forensic investigators may effectively address permitted modification of digital evidence, while traditional hash techniques are useless in this scenario. The suggested framework's fundamental process is shown in Figure 1. Each stage will be discussed in depth.

The efficiency of the proposed system has been verified for application in the field of image forensics. Only images are used in the paper. However, this is a universal approach for different types of data such as audio, video, image, and files. The reasons for choosing images in the application lie in the following factors: (1) a large number of cases within the scope of the work of digital forensics experts are related to image counterfeiting as they represent the main segment in transactions for individuals, such as images of signatures and checks; (2) with the advancement and availability of powerful image processing software tools and computer technology, it is very easy to manipulate digital images. So, it is essential to determine the authenticity, integrity, reliability, and origin of digital images; (3) images can be used in very important fields such as forensic science, medicine, astronomy, and surveillance.

The investigator does not modify the evidence, but the evidence may be altered by benign modification within some application such as compression. The pseudo-randomness of cryptographic hash algorithms makes it hard to identify similar files even if one bit of the input is changed. A hash function that does not retain the resemblance of files (e.g., different versions of a file) is necessary in the area of computer forensics. How forensic investigators may use traces from such situations is becoming more difficult to determine.

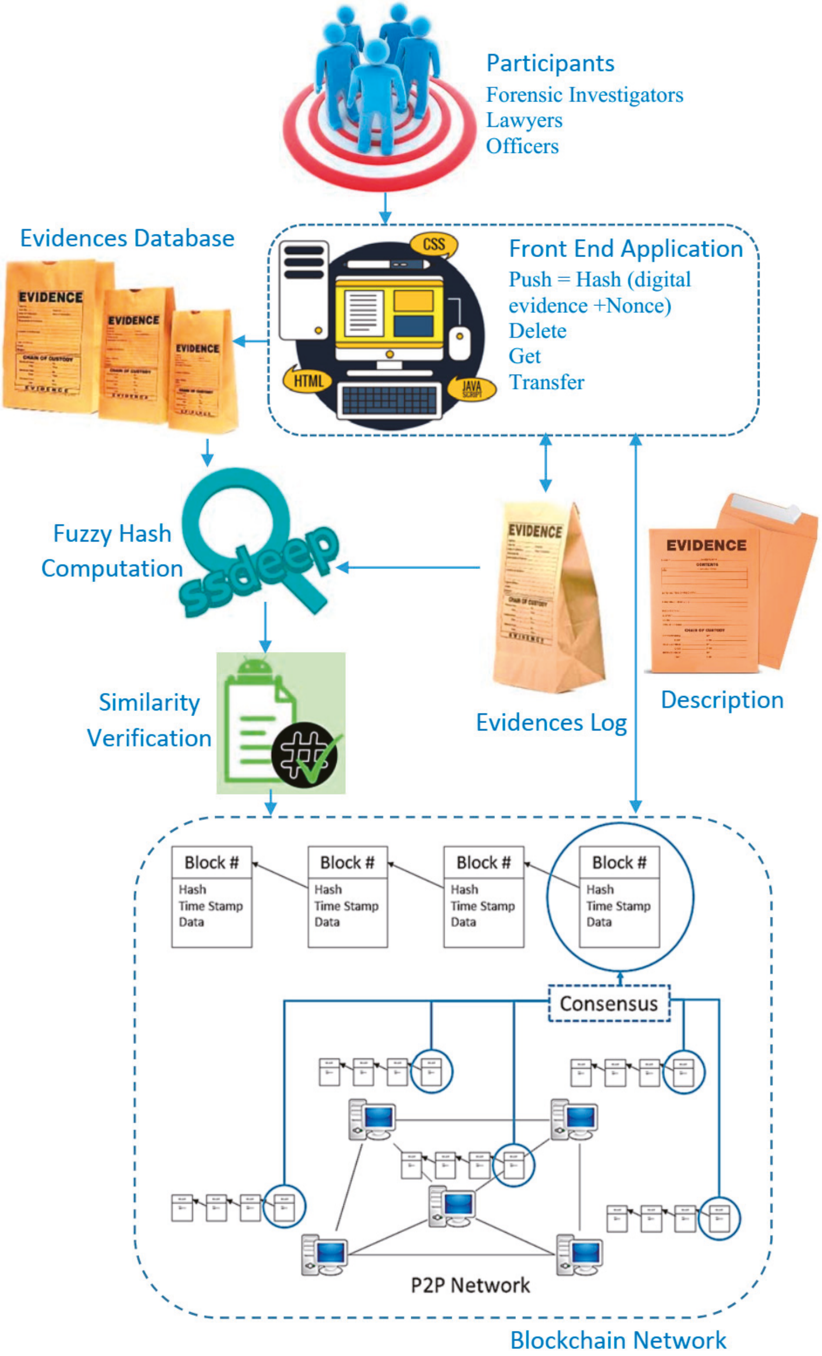


Figure 1. Proposed framework for protecting digital evidence integrity under uncertainty.

3.1. Participants

Authorized parties (forensic investigation) gather digital evidence (images) and then register it in blockchain. Lawyers, the police, the defense, and the court all participate in forensic investigations because they need information regarding the CoC at various points throughout an investigation. Only authorized parties have access to the data associated with a specific piece of evidence [6]. Each authorized entity has a unique identity that is publicly known, and he or she possesses credentials that enable authentication and action throughout the CoC process [21].

3.1.1. Front End

This part is intended to serve as an interface for authorization, to access permissions, and for media. It allows for the downloading of digital evidence and certificates of authenticity, in line with access permissions and levels. The Blockchain interface enables participants to see, invoke, and query blocks, transactions, and chain codes [15,20]. The front end produces a hash of the digital evidence and a nonce that uniquely identifies it (Evidence ID). As the hash generates the ID and the value nonce is randomly selected to guarantee the uniqueness of the evidence's identification, it aids in preserving the integrity of digital evidence throughout its lifetime [21].

3.1.2. Evidence Log

The evidence log keeps track of user interactions with digital evidence. This Evidence Log is implemented on the blockchain and contains information on each piece of evidence on which decision-making depends, including its ID, a description, the submitter's (creator's) identity, and the full history of owners up to the present one, including the dates of ownership transfers. The evidence log is built on top of a peer-to-peer network that includes all authorized entities. A network of this kind may be split into two distinct groups of nodes [15,21]: (1) validator nodes: they are primarily responsible for maintaining a copy of the blockchain; validating transactions; and creating, proposing, and adding blocks to the chain (i.e., participate in the consensus protocol). (2) Lightweight nodes: they are considered clients of the chain since they just issue transactions and depend on validators to add and validate them.

3.2. Blockchain

A blockchain is a decentralized ledger that is maintained by trustless nodes in a peer-to-peer network. Data are stored on the blockchain in blocks that are linked through a connection to the hash value of each block. It is not feasible to modify data in the midst of a block [15]. The first responder initiates forensic-chain by hashing digital evidence (image) and securely storing it on the blockchain through the smart contract. Additional information such as the time and date of the incident, the location of the crime scene, the address to which evidence is transferred, and the present condition of the evidence are also stored on the blockchain. The chain of custody for digital forensics on the blockchain has the potential to significantly improve forensic applications by ensuring the integrity and security of digital evidence while achieving the intended result [9].

As there are just a few peers connected to the network, the block size on the blockchain is smaller. In contrast, not all nodes are required to download the whole blockchain in order to be operational. Some members in the blockchain network participate just for the purpose of making transactions and not for the purpose of verifying them. Full nodes are a subset of nodes that fall into this category. Participation in the current transaction requires the use of complete nodes. The block headers and transactions in each block must be downloaded in their entirety, which implies users must download the whole blockchain's contents. With the ever-increasing size of the blockchain, scalability also becomes a problem. Furthermore, the blockchains' number is determined by the number of available digital pieces of evidence.

The proposed system depends on the piecewise hashing technology for cryptography since the main contribution is to handle uncertainty in CoC. Piecewise hashing uses an arbitrary hashing algorithm to create many checksums for a file instead of just one. Rather than generating a single hash for the entire file, a hash is generated for many discrete fixed-size segments of the file. The following characteristics describe this particular kind of hashing: (1) a hash function should be computationally difficult to reverse "pre-image resistance"; (2) it should be difficult to discover another input with the same hash if you know the hash of the input you're looking for "second Pre-Image Resistance"; (3) it should be difficult to locate two inputs of the same length that have the same hash value if this characteristic is present "collision-free hash function" [22–24].

3.2.1. Piecewise Hashing

To account for the uncertainty associated with evidence item changes, we utilized Fuzzy Hashing (FH) rather than conventional hashes such as SHA 256 in our study. FH, also known as Context-Triggered Piecewise Hashing (CTPH), is a mix of Piecewise and Rolling Hashing (RH). Unlike traditional hashes, where their hashes (checksums) can be interpreted as correct or incorrect, and as black or white, CTPH is more akin to the "grey hash type" as it can identify two files that are likely near duplicates of one another but would not be detected using traditional hashing methods [23]. RH generates 'segments' of conventional hash strings by generating a pseudo-random value depending on the context of the input. In comparison, PH (Piecewise Hashes), such as conventional hashes, produce a final checksum for the whole picture. They circumvent the latter's restrictions by segmenting the whole image into defined segments and then generating hash values for each of these parts. Finally, the produced values comprise the final hash sequence. FH employs the concept of PH to preserve data similarity in this study. Additionally, PH was designed to minimize possible mistakes during forensic imaging, ensuring that the data's integrity is absolute and complete since only one hash segment is void [23,24].

3.2.2. Approximate Matching

Approximate matching is an exciting new technique for determining the similarities between two digital objects. Numerous approximation matching techniques developed to address contemporary issues in digital forensics are essentially based on the capacity to describe objects as sets of characteristics, which simplifies the similarity problem by limiting it to the well-defined domain of set operations [25]. There are eight well-known approximation matching algorithms, including the following ssdeep, sdhash, mrsh-v2, bbHash, mvHash-B, SimHash, saHash, and TLSH. While the first three algorithms remain expanded and relevant, the last four algorithms are less promising in terms of digital forensics for a variety of reasons, including recall and accuracy rates, runtime efficiency, and detection capabilities. For cross-correlation, the final method (TLSH) is less powerful than sdhash and mrsh-v2 [25]. While ssdeep is the most well-known CTPH use today, another method, Multi-Resolution Similarity Hashing, version 2 (MRSH-V2), has been suggested based on the same principles or enhancements to the original ssdeep algorithm [26].

In ssdeep, the system computes the similarity of two files based on their signatures throughout the comparison process [26]. Ssdeep analyzes two strings and calculates the least number of operations required to convert one string into the other using an edit distance method based on Levenshtein distance. While ssdeep is very efficient at detecting similarities between text files, it has a poor detection rate for images due to the possibility of an active adversary exploiting it [23]. In comparison, Sdhash (Similarity Digest hash) encodes the output hash features with a low empirical probability using Bloom Filters. Its results are based on a "similarity score calculated by computing the normalized entropy of the digests, which runs from 0 to 100, with 0 being a mismatch and 100 representing a perfect or near match. The sole drawback discovered for sdhash was its execution time [23].

Mrsh-v2 overcomes ssdeep's limitations and becomes quicker than sdhash [25]. The main objective of MRSH-v2 is to compress and produce a similarity digest for every byte sequence. Similarity digests are constructed in such a manner that they may be compared to one another, generating a similarity score. Each digest of similarity is composed of Bloom filters. To generate the similarity digest, MRSH-v2 divides the input into roughly 160-byte pieces (sub hashes). These chunks are hashed using FNV (Fowler/Noll/Vo) Fast non-cryptographic hash function to establish the Bloom filter's five bits. To chunk the input, it employs a seven-byte window that glides across it byte by byte. Approximate matching is accomplished by comparing similarity digests. A pairwise comparison of two file sets is needed to compare them [27,28].

The root node of a hierarchical Bloom filter tree is a Bloom filter that represents the whole collection. When searching for an image, if a match is discovered at the root of the tree, all of the tree's child nodes may be searched. The method of determining if a file matches a Bloom filter node is identical to that of adding a file to the tree. Rather than putting each hash into the node, the sub hashes are compared to the Bloom filter to see whether they are included inside it. If a node has a certain number of consecutive hashes, it is considered a match [27].

3.2.3. Similarity

A similarity tool's ultimate aim is to function as a drop-in substitute for the crypto hashes used in forensic file practice for file filtering [28]. Approximate matching may be accomplished using two distinct abstractions: byte wise matching and semantic matching. (1) Byte wise matching: this algorithm works at the byte level and accepts only byte sequences as input. Byte wise algorithms serve two primary purposes. A feature extraction function detects and extracts properties from objects in order to compress them for comparative purposes. Then, a similarity function compares these compressed versions in order to provide a normalized match score. Typically, this comparison is made using string formulae such as Hamming and Levenshtein distances [25]. Byte-wise has a number of restrictions, including [25]. (1) It is unable to detect similarities at a higher level of abstraction, for example, semantically. (2) It is unable to properly match two image files that contain the same semantic image but are stored in various file kinds and formats as a result of their differing binary encodings. (3) Due to the absence of a universally accepted definition of similarity, not all types of byte-level similarity are equally useful since certain artifacts (e.g., headers and footers) are trivial and result in false positives.

This research focuses on the second type, (2) semantic matching, which operates on the content visual layer (i.e., digital evidence images) and thus closely resembles human behavior, for example, the similarity of the content of a JPG and a PNG image, despite the fact that the image file types are different. To put it another way, two images are semantically similar if they convey the same information. For instance, a JPG file is semantically equivalent to an exported PNG file containing the same image. Their cryptographic hashes will not be same, but the images will be identical [25]. To compare two hash values, a comparison function is required. The comparison function takes two hash values as input and returns a number between 0 and X, where X is the maximum match score. A score of X indicates that the hash values are identical or nearly similar, implying that the input files are also identical or nearly identical. The similarity score should ideally be between 0 and 100 and expressed as a percentage.

The suggested framework uses MRSH-v2 for creating the digital grey hash for each block within the blockchain network that utilizes the Hierarchical Bloom Filter Tree (HBFT) approach. As stated in [27], HBFT is quite good at detecting files that share at least 40% of their content, and it has excellent recall when dealing with identical sets of data. This means that the HBFT data structure is an effective alternative to all-against-all comparisons while also delivering significant speed benefits. The HBFT approach yielded a recall level of 95% for similar files when using mrsh-v2 as ground truth. Therefore, the proposed framework has considered 95% as the appropriate metric for resemblance [28]. See [29–34] for more information regarding fuzzy hashing techniques.

3.3. Peer to Peer (P2P) Network

P2P is used to create the network architecture and to facilitate communication between the blockchain layer and the rest of the network (responsible for constructing a blockchain for each node in the underlying network). The majority of blockchain schemes use a peer-to-peer network as a blockchain network. This work utilizes a peer-to-peer network to organize nodes, offers peer-to-peer routing, secures the transfer of proof information, and maintains the Blockchain's consensus. Existing peer-to-peer network methods may be utilized directly or modified to build the Blockchain's network [10].

3.4. Consensus Mechanism

The blockchain consensus process selects a node to generate and broadcast the blockchain next block and ensures that each node's blockchain is consistent [10]. A blockchain transaction is verified via the application of a consensus concept. Consensus ensures that each transaction has its own independent witness mechanism. On the blockchain, there are many forms of consensus, including Proof of Work (PoW), Proof of Stack (PoS), and Proof of Authority (PoA). Consensus types vary according to how the blockchain interacts with data storage [15].

With PoW, nodes compete against one another by solving a mathematical problem to confirm transactions and create new blocks. While solving a block is a computationally demanding job, validating it is straightforward. To further incentivize such a system, solving a block also results in the mining of a certain number of bitcoins, which serves as the incentive for block makers (often referred to as miners) [21]. PoW is suitable for permissionless networks, that is, networks in which nodes may join without prior authorization. The primary disadvantage of PoW is its high energy consumption, which also precludes its use in some situations [21]. This has resulted in the study of other types of blockchain consensus, such as PoA. This study focuses on PoA, which is usually used in permissioned networks, i.e., networks in which nodes cannot join and become validators freely. With the PoA, validators must be pre-authorized and their identities must be known. As a consequence, behaving maliciously leads in a loss of personal reputation and, eventually, expulsion from the validator set [21].

3.5. Hyperledger Blockchain Platform

Hyperledger Fabric (HLF) is a blockchain-based system for electronic digital record exchange across several organizations. Recently, several blockchain systems have been created by different businesses, including Ethereum, Corda, and Ripple [35]. The Hyperledger Composer (HLC) is a framework for building blockchain applications that significantly speeds up and simplifies the process of designing blockchain use cases. One of the many benefits of HLC is that it is completely open-source, with an open governance architecture that allows for contributions by anybody [6]. By design, HLC satisfies all of the criteria for developing an automated system that is both robust and secure in its recording of all the information related to the evidence collection process for a specific cyber forensic case. HLC is compatible with and runs on top of the current HLF blockchain architecture and runtime, enabling pluggable blockchain consensus protocols to guarantee that transactions are verified according to the policy established by the designated business network members [6].

The proposed framework in this article is based on HLF and HLC and offers the following major benefits [6,36]: (1) it is distinguished from the others by its usage of the permissioned blockchain idea, in which transaction processing is delegated to a select group of trustworthy network members; (2) as a consequence, the resulting environment is more regulated and predictable than public permissionless blockchains; (3) block generation does not require resource-intensive computations associated with PoW techniques; (4) due to its modular nature, it enables the employment of a variety of methods to achieve agreement among business process participants; and (5) Ethereum is probably not the ideal cryptocurrency to use for crime-scene investigation. Digital forensic investigations require confidentiality and are conducted by genuine and trustworthy parties.

From a functional standpoint, the HLF network's nodes are classified as follows [36]: (1) clients initiate transactions, participate in their processing, and broadcast transactions to ordering services; (2) peers execute the transaction processing workflow, verify them, and maintain the blockchain registry; the blockchain registry is an append-only data structure that contains a hash chain of all transactions, as well as a concise representation of the latest ledger state; (3) Ordering Service Nodes (OSN) or, simply, orders establish the general order of all transactions in the blockchain using the distributed consensus algorithm; each transaction contains updates to the system's state, the history of which is stored in the blockchain, and cryptographic signatures of endorsing peers; the separation of processing nodes (peers) and transaction order keeps HLF's consensus as modular as feasible and facilitates protocol replacement.

To define business processes within the framework of the (HLF and HLC) platform, a variety of concepts are employed, the most important of which are assets, participants, and network-stored transactions. (1) Assets: anything of value that can be traded or shared via a network is considered an asset. The suggested approach treats digital evidence and the comprehensive information associated with it as an asset that is kept in HLC's asset registry. (2) Collaborators: participants in the forensic chain model are forensic investigators. In HLC, the participant's structure is represented using a file. It is possible to generate new instances of the modeled participant and add them to the participant register.

Additionally, HLC needs blockchain IDs as a form of identification, and an identity registry stores a collection of mappings between identities and participants. At any point in time, admin peers controlled by companies in the hyperledger composer blockchain consortium may add new participants with suitable identity responsibilities to address a specific scenario. Participants may exchange information in a secure manner using the channels available on the (HLF and HLC) platform. (3) Transactions are used to explain the activities that participants may take on assets as they travel through the network. Transactions in the proposed framework either record information about the evidence or the evidence transfer event on the network. See [37–40] for more information regarding hyperledger blockchain platform.

3.6. Evidence Database

The evidence database is a standard database and/or file repository that stores the actual digital evidence together with an identification ID computed from the evidence's hash and a nonce. This database is disseminated and is maintained by a number of reputable organizations (e.g., law, court, officers). Additionally, each access is granted only if the asking organization is allowed to provide it in accordance with its function. There are two reasons for this split (between the Evidence Log and the Evidences database). To begin, evidence may be too big to be kept effectively on the blockchain (for example, a piece of evidence may be a bit-by-bit copy of a storage device of several TBs of capacity). Second, and most crucially, if pieces of evidence are kept on the blockchain, they are accessible to all nodes in the blockchain network, while only authorized nodes should be permitted to collect evidence. As a result, we keep just information on the CoC process and a hash of the evidence in the blockchain, which enables us to check the integrity of pieces of evidence throughout acquisition [21]. See [41–47] for more information about protecting digital evidence integrity and preserving chain of custody.

4. Performance Evaluation and Analysis

Performance is the most desired characteristic of any problem-solving endeavor, and this is also true for Blockchain-based solutions. We utilized Hyperledger Caliper to assess the performance of our prototype. Caliper enables users to benchmark the performance of various blockchain systems against a specified set of use cases and produce reports that include performance metrics such as transactions per second (tps) and transaction latency (the time elapsed from the issue of the transaction to its inclusion in the blockchain). The experiments were conducted on an Intel Core i7–5500U, 2.4 GHz processor, 8 GB DDR3 RAM laptop, and Windows 10 operating system. The code was written in Python language using Python 3.6 software.

4.1. Performance Analysis

The first set of experiments was implemented to test our prototype using Caliper's 2-organization-1-peer and 3-organization-1-peer network models with four clients in the first round of tests. To ascertain our suggested framework's transactional efficiency, we created a test file that targeted two primary functionalities of our framework, namely, evidence creation and evidence transfer, due to their direct participation in changing the Blockchain state. We conducted ten rounds of testing with varying transaction volumes and send transaction rates. Multiple runs of the test were required to get the average values of performance indicators with a low chance of error. Tables 1 and 2 show the latency and throughput for various rounds of 2-organization-1-peer and 3-organization-1-peer network architectures. The performance assessment results indicate that the prototype's throughput achieves a maximum value and then begins to decrease as the transmit rate increases. The highest throughputs obtained in 2-organization-1-peer and 3-organization-1-peer network architectures are 15 tps and 10 tps, respectively. Additionally, the results indicate that increasing the number of peers reduces the prototype's throughput, which is consistent with the characteristic of Hyperledger-based consortium Blockchains.

Table 1. Performance evaluation results with 2-organization-1-peer network mode.

Round	Send Rate (tps)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (tps)
1	6	0.85	0.70	0.77	5
2	11	1.18	0.74	0.98	9
3	16	1.46	0.49	1.13	13
4	21	2.89	0.61	1.93	14
5	26	4.06	0.84	2.72	14
6	30	5.80	1.05	4.37	15
7	35	7.27	1.32	5.76	15
8	40	21.61	8.36	16.15	8
9	43	11.49	2.49	8.38	15
10	49	13.88	8.57	11.85	13

Table 2. Performance evaluation results with 3-organization-1-peer network framework.

Round	Send Rate (tps)	Max Latency (S)	Min Latency (S)	Avg Latency (S)	Throughput (tps)
1	6	1.24	1.01	1.16	5
2	11	8.32	2.74	6.34	4
3	16	4.60	1.00	3.13	8
4	21	8.42	5.24	7.01	8
5	26	9.56	3.95	7.11	10
6	30	11.62	3.85	9.07	10
7	33	14.16	3.22	10.99	10
8	39	17.16	10.77	14.34	9
9	46	47.84	19.93	34.37	5
10	50	19.35	12.21	16.29	10

The second round of tests assessed the block generator's load, which is used to determine the distribution of blocks generated by each node. This shows if each node in the blockchain network being used has an equal probability of producing blocks. We utilized a 1000-node architecture in the simulator and created 105 blocks sequentially, counting the blocks generated by each node. The cumulative percentage of produced blocks containing x nodes is shown in Figure 2, where k is the number of node names. The more evenly distributed the load, the more likely the line will be straight. When k equals one, the curve exhibits a sharp rise. The demand on the generator is balanced evenly by increasing the number of node names. The greater the number of node names, the more linear the growth becomes. However, as the number of node names grows, load balancing's growth impact progressively diminishes. By adding a modest number of node names, these block generators may significantly improve load balancing. The number of blocks produced is centered on the mean. In general, when *k* equals 5, the load balancing impact is satisfactory for the block generator.

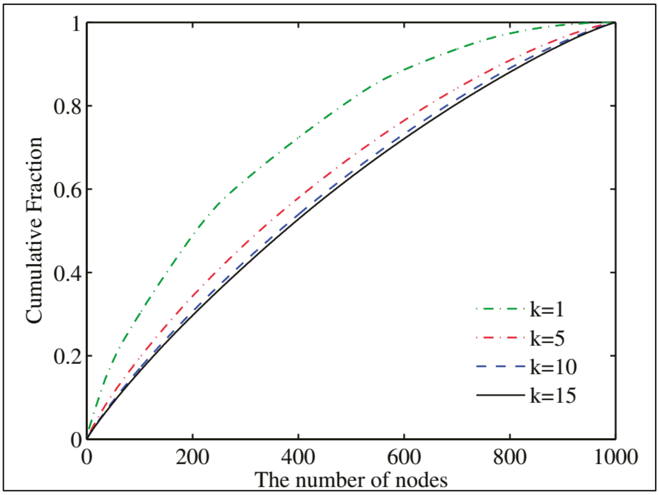


Figure 2. The cumulative distribution of generated blocks.

The third set of experiments was conducted to evaluate the size of the blockchain against different numbers of blocks on a topology with 1000 nodes. The name number was set to one and the group size variable, *h*, was set to three bits for the topology. A block could contain no more than 2000 transactions. Following that, we determined the blockchain's storage capacity on each node. We were primarily concerned with the distribution of full blocks (block headers and contents) and the blockchain's size. The distribution of full blocks stored by each node represented the blockchain's load balancing mechanism. We conducted the experiment three times. Each time, we adjusted the variable *h* to create a new group size and then counted the number of full blocks stored in each node. Figure 3 illustrates the blockchain's size as a function of the block count. The maximum, mean, and minimum blockchain sizes are all determined using the mixed blockchain, whereas the entire blockchain size is determined using a typical scenario in which all nodes hold the whole blockchain. The mixed blockchain is much smaller than the regular blockchain. For all four kinds of outcomes, the blockchain's size grows linearly as the number of blocks increases, which is consistent with the theoretical theory.

We conducted the last set of experiments to determine the time required to conduct a full search, in comparison to MRSH-v2, and to determine the approach's success in locating the 100 "illegal" files included verbatim in the hard disk image, as well as the 40 files from the image that are similar to "illegal" files, as defined by MRSH-v2. A collection of simulated "known-illegal" images consisted of 4000 images plus 140 more images, as follows: within the 4000 "illegal" images, there were 100 images; 40 images were not included in the "illegal" images but showed a high degree of resemblance to images in the corpus, as determined by MRSH-v2.

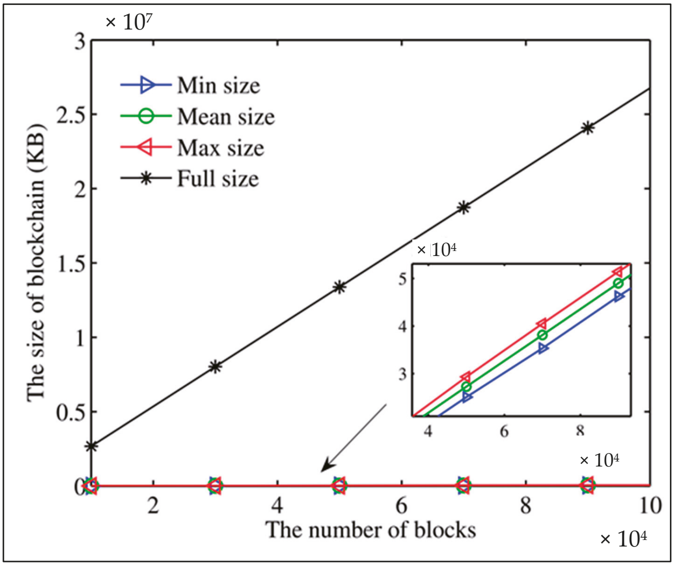


Figure 3. The blockchain's size as a function of the block count.

The main measure was the time needed to execute the whole process, which included the time required to construct the tree, search the tree, and perform pairwise comparisons at the leaves. MRSH-v2 ran for a total of 2592 s. Figure 4 illustrates the running times. The tree was constructed using the smaller sample of 4000 "illegal" images, and then searches were performed for all of the images in the bigger corpus. The "Search Time" column covers both the time spent searching the tree and the time spent doing leaf comparisons. As anticipated, having more leaf nodes resulted in the quickest execution time. The entire duration of the race was 1182 s (a 54% reduction in the time required for an all-against-all pairwise comparison). Due to the paired approach's lack of scalability, this discrepancy is expected to be much more apparent with bigger datasets.

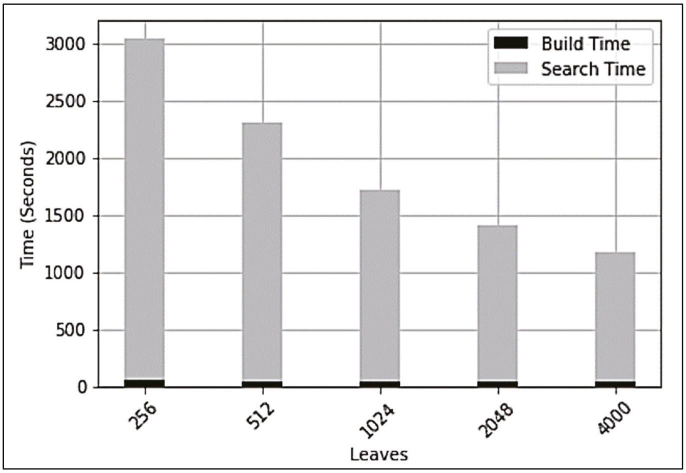


Figure 4. Time to search for planted evidence (including pairwise comparisons).

4.2. Analysis of Possible Attacks

As far as forensics are concerned, both blacklisting and whitelisting attacks are discussed in this section. Anti-blacklisting/anti-whitelisting may be used to conceal information from the perspective of an attacker. An active attacker manipulates a file such that fuzzy hashing does not recognize the files as being identical, which is what is meant by "anti-blacklisting." If a human observer cannot tell the difference between the original and the manipulated version, we consider the attack to be effective. If a file was successfully modified, it would be labelled as an unknown file rather than a known-to-be-bad file. This anti-blacklisting attack aims to alter a single byte inside each chunk while keeping track of the exact locations of the trigger points. Change the triggering such that the extent of each change is determined by the Hamming distance, which is the most apparent concept. As stated in [33], in a worst-case scenario, each building block has a Hamming distance and a 'one-bit change' is enough to manipulate the triggering. In this case, an active adversary approximately needs to change one bit for each position. Actually, a lot of 100 more changes needs to be done as there are also positions where the Hamming distance has small distance.

For anti-whitelisting to work, the attacker must utilize a hash value from one of the files on the whitelist in order to change another file (typically one of the bad ones) such that the new file's hash value is identical to one on the whitelist. An attack is deemed effective if a human observer cannot detect any differences between the original and altered versions. Since files may be created for a given signature by generating legal trigger sequences for each building block and inserting zero-strings in between, this technique is not considered preimage-resistant. Even though it should be feasible, changing the hash value of a particular file will lead to a worthless file. An active adversary's initial action is to delete all currently active trigger sequences. As a second step, he must completely mimic the white-listed file's triggering behavior, which will result in many additional modifications to the system.

5. Conclusions

The integrity and credibility of the digital evidence in a single process for managing the chain of custody are critical components of these operations (or chain of evidence). The purpose of this study is to determine the efficacy of fuzzy hashing algorithms inside blockchain technology, as opposed to conventional cryptographic hash algorithms, in preserving the integrity of digital evidence in image forensics for assessing similarities. We developed and tested a prototype of a forensic chain model based on a hyperledger composer. According to the performance evaluation, fuzzy hash-based blockchains proved to be an effective support for the chain of custody process due to their ability to sustain a realistic workload with a manageable overhead in terms of memory used to store the chain and their ability to handle the chain of custody-related uncertainty. Future work includes testing the efficiency of the suggested framework when handling a large number of digital pieces of evidence.

Author Contributions: Conceptualization, S.D., H.E. and S.M.; methodology, M.A., A.I. and H.E.; software, M.A., A.I. and H.E.; validation, S.D., H.E. and S.M.; formal analysis, S.D., M.A., A.I. and H.E.; investigation, S.D., M.A. and A.I.; resources, S.D., H.E. and S.M.; data curation, M.A., A.I. and H.E.; writing—Original draft preparation, S.D., M.A., A.I. and H.E.; writing—Review and editing, S.D. and S.M., visualization, M.A., A.I. and H.E.; supervision, S.D. and S.M.; project administration, M.A., A.I. and H.E.; funding acquisition, M.A., A.I. and H.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dosisa, S.; Homema, I.; Popova, O. Semantic representation and integration of digital evidence. *Procedia Comput. Sci.* **2013**, *22*, 1266–1275. [CrossRef]
2. Prayudi, Y.; Azhari, S. Digital chain of custody: State of the art. *Int. J. Comput. Appl.* **2015**, *114*, 1–9. [CrossRef]
3. Campbell, N.; Goodyear, T.; Messer, W.; Stuart, E.; Fairbanks, J. Digital witness: Remote method for volunteering digital evidence on mobile devices. In Proceedings of the IEEE International Conference on Technologies for Homeland Security, Woburn, MA, USA, 23–24 October 2018; pp. 1–5.
4. Patel, J.; Bhatt, N. Review of digital image forgery detection. *Int. J. Recent Innov. Trends Comput. Commun.* **2017**, *5*, 152–155.
5. Varkey, A.; Nair, L. Robust image forgery detection and classification in copy-move using SVM. *Int. J. Adv. Res. Trends Eng. Technol.* **2018**, *5*, 89–93.
6. Hamid, A.; Naaz, R. Forensic-chain: Blockchain based digital forensics chain of custody with POC in hyperledger composer. *Int. J. Digit. Investig.* **2019**, *28*, 44–55.
7. Sadiku, M.; Shadare, A.; Musa, S. Digital chain of custody. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2017**, *7*, 117–118. [CrossRef]
8. Ćosić, J.; Baca, M. (Im) Proving chain of custody and digital evidence integrity with time stamp. In Proceedings of the IEEE International Conference on Convention, Opatija, Croatia, 21–23 May 2010; pp. 1226–1230.
9. Auqib, L.; Roohie, M. Forensic-chain: Ethereum blockchain based digital forensics chain of custody. *Sci. Pract. Cyber Secur. J.* **2017**, *1*, 21–27.
10. Tian, Z.; Li, M.; Qiu, M.; Sun, Y.; Su, S. Block-DEF: A secure digital evidence framework using blockchain. *Int. J. Inf. Sci.* **2019**, *491*, 151–165. [CrossRef]
11. Ćosić, J.; Ćosić, Z.; Baca, M. An ontological approach to study and manage digital chain of custody of digital evidence. *Int. J. Inf. Organ. Sci.* **2011**, *35*, 1–13.
12. Saleem, S.; Popov, O.; Dahman, R. Evaluation of security methods for ensuring the integrity of digital evidence. In Proceedings of the IEEE International Conference on Innovations in Information Technology, Abu Dhabi, United Arab Emirates, 25–27 April 2011; pp. 220–225.
13. Rasjid, Z.; Soewito, B.; Witjaksono, G.; Edi, A. A review of collisions in cryptographic hash function used in digital forensic tools. *Procedia Comput. Sci.* **2017**, *116*, 382–392. [CrossRef]
14. Korus, P. Digital image integrity—A survey of protection and verification techniques. *Digit. Signal Process.* **2017**, *71*, 1–26. [CrossRef]
15. Yunianto, E.; Prayudi, Y.; Sugiantoro, B. B-DEC: Digital evidence cabinet based on blockchain for evidence management. *Int. J. Comput. Appl.* **2019**, *181*, 22–29.
16. Cosic, J.; Cosic, Z. Chain of custody and life cycle of digital evidence. *Comput. Technol. Appl.* **2012**, *3*, 126–129.
17. Giova, G. Improving chain of custody in forensic investigation of electronic digital systems. *Int. J. Comput. Sci. Netw. Secur.* **2011**, *11*, 1–9.
18. Gayed, T.; Lounis, H.; Bari, M. Cyber forensics: Representing and (im) proving the chain of custody using the semantic web. In Proceedings of the IEEE International Conference on Advanced Cognitive Technologies and Applications, Nice, France, 25–29 October 2012; pp. 19–23.
19. Widatama, K.; Prayudi, Y.; Sugiantoro, B. Application of RC4 cryptography method to support xml security on digital chain of custody data storage. *Int. J. Cyber-Secur. Digit. Forensics* **2018**, *7*, 230–237. [CrossRef]
20. Brotsis, S.; Kolokotronis, N.; Limniotis, K.; Shiaeles, S.; Kavallieros, D.; Bellini, E.; Pavué, C. Blockchain solutions for forensic evidence preservation in IoT environments. In Proceedings of the IEEE International Conference on Network Softwarization, Paris, France, 24–28 June 2019; pp. 110–114.
21. Bonomi, S.; Casini, M.; Ciccotell, C. B-CoC: A blockchain-based chain of custody for evidences management in digital forensics. *arXiv* **2018**, arXiv:1807.10359, 1–18.
22. Baqir, M.; Saleem, S.; Zulqarnain, R. Protecting digital evidence integrity and preserving chain of custody. *J. Digit. Forensics Secur. Law* **2017**, *12*, 121–130.
23. Sarantinos, N.; Benzaıd, C.; Arabiat, O.; Al-Nemrat, A. Forensic malware analysis: The value of fuzzy hashing algorithms in identifying similarities. In Proceedings of the IEEE International Conference Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; pp. 1783–1787.
24. Dodson, J.; Siraj, A. Applying fuzzy hashing to steganography. *Int. J. Future Comput. Commun.* **2015**, *4*, 421–425. [CrossRef]
25. Vikram, S.; Breitinger, F.; Baggili, I. Bytewise approximate matching: The good, the bad, and the unknown. *J. Digit. Forensics Secur. Law* **2016**, *11*, 59–78.
26. Martinez, V.; Álvarez, F.; Encinas, L. State of the art in similarity preserving hashing functions. In Proceedings of the IEEE International Conference on Security and Management, Washington, DC, USA, 12–14 May 2014; pp. 1–7.

27. Lillis, D.; Breitinger, F.; Scanlon, M. Expediting MRSH-v2 approximate matching with hierarchical bloom filter trees. In Proceedings of the IEEE International Conference on Digital Forensics and Cyber Crime, Cham, Switzerland, 9–11 October 2017; pp. 144–157.
28. Breitinger, F.; Baier, H. A fuzzy hashing approach based on random sequences and hamming distance. In Proceedings of the IEEE International Conference on Digital Forensics, Security and Law, Darmstadt, Germany, 2–5 December 2012; pp. 89–100.
29. Naik, N.; Jenkins, P.; Savage, N.; Yang, L. Cyberthreat Hunting—Part 2: Tracking ransomware threat actors using Fuzzy Hashing and fuzzy c-means clustering. In Proceedings of the IEEE International Conference on Fuzzy Systems, New Orleans, LA, USA, 23–26 June 2019; pp. 1–6.
30. Naik, N.; Jenkins, P.; Savage, N.; Yang, L.; Boongoen, T. Fuzzy-Import Hashing: A malware analysis approach. In Proceedings of the IEEE International Conference on Fuzzy Systems, Glasgow, UK, 19–24 July 2020; pp. 1–8.
31. Lu, H.; Zhang, M.; Xu, X.; Li, Y.; Tao Shen, H. Deep fuzzy hashing network for efficient image retrieval. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 1–11. [CrossRef]
32. Naik, N.; Jenkins, P.; Savage, N. A Ransomware detection method using fuzzy hashing for mitigating the risk of occlusion of information systems. In Proceedings of the IEEE International Conference on Systems Engineering, Edinburgh, UK, 1–3 October 2019; pp. 1–6.
33. Naik, N.; Jenkins, P.; Savage, N.; Yang, L.; Asagar, K. Fuzzy hashing aided enhanced yara rules for malware triaging. In Proceedings of the IEEE International Conference on Computational Intelligence, Canberra, ACT, Australia, 1–4 December 2020; pp. 1138–1145.
34. Naik, N.; Jenkins, P.; Gillett, J.; Mouratidis, H.; Naik, K. Lockout-tagout ransomware: A detection method for ransomware using fuzzy hashing and clustering. In Proceedings of the IEEE International Conference on Computational Intelligence, Xiamen, China, 6–9 December 2019; pp. 2792–2796.
35. Wutthikarn, R.; Guang, Y. Prototype of blockchain in dental care service application based on hyperledger composer in hyperledger fabric framework. In Proceedings of the IEEE International Conference on Computer Science and Engineering, Chiang Mai, Thailand, 21–24 November 2018; pp. 1–4.
36. Demichev, A.; Kryukov, A.; Prikhodko, N. The approach to managing provenance metadata and data access rights in distributed storage using the hyperledger blockchain platform. In Proceedings of the IEEE International Conference on Ivannikov Ispras Open Conference, Moscow, Russia, 22–23 November 2018; pp. 131–136.
37. Choi, W.; Won-Ki Hong, J. Performance Evaluation of Ethereum private and test net networks using hyperledger caliper. In Proceedings of the IEEE International Conference on Asia-Pacific Network Operations and Management Symposium, Tainan, Taiwan, 8–10 September 2021; pp. 325–329.
38. Ampel, B.; Patton, M.; Chen, H. Performance modeling of hyperledger sawtooth blockchain. In Proceedings of the IEEE International Conference on Intelligence and Security Informatics, Shenzhen, China, 1–3 July 2019; pp. 59–61.
39. Sukhwani, H.; Wang, N.; Trivedi, K.; Rindos, A. Performance modeling of hyperledger fabric (permissioned blockchain network). In Proceedings of the IEEE International Conference on Network Computing and Applications, Cambridge, MA, USA, 1–3 November 2018; pp. 1–10.
40. Park, I.; Lee, T.; Jang, J. Trade-off in implementation of p2p energy trading over hyperledger blockchain. In Proceedings of the IEEE International Conference on Consumer Electronics—Asia, Seoul, Korea, 1–3 November 2020; pp. 1–4.
41. Frankowski, A.; Dębski, A. Recovery of forensic traces with use of state-of-the-art imaging techniques—system for marking, tracing and maintaining chain of custody. *Issues Forensic Sci.* **2018**, *299*, 52–56. [CrossRef]
42. Al-Khateeb, H.; Epiphaniou, G.; Daly, H. Blockchain for modern digital forensics: The chain-of-custody as a distributed ledger. In *Blockchain and Clinical Trial*; Springer: Cham, Switzerland, 2019; pp. 149–168.
43. Burri, X.; Casey, E.; Bolle, T.; Jaquet-Chiffelle, D. Chronological independently verifiable electronic chain of custody ledger using blockchain technology. *Forensic Sci. Int. Digit. Investig.* **2020**, *33*, 300976. [CrossRef]
44. Tanner, A.; Bruno, J. Timely: A Chain of Custody Data Visualizer. In Proceedings of the IEEE Southeast Conference, Huntsville, AL, USA, 11–14 April 2019; pp. 1–5.
45. Dasaklis, T.; Casino, F.; Patsakis, C. Sok: Blockchain solutions for forensics. In *Technology Development for Security Practitioners*; Springer: Cham, Switzerland, 2021; pp. 21–40.
46. Chopade, M.; Khan, S.; Shaikh, U.; Pawar, R. Digital forensics: Maintaining chain of custody using blockchain. In Proceedings of the Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 12–14 December 2019; pp. 744–747.
47. Fontani, M.; Bianchi, T.; De Rosa, A.; Piva, A.; Barni, M. A forensic tool for investigating image forgeries. *Int. J. Digit. Crime Forensics* **2013**, *5*, 15–33. [CrossRef]

symmetry

MDPI

Article

A Software Reliability Model with Dependent Failure and Optimal Release Time

Youn Su Kim [1], Kwang Yoon Song [1], Hoang Pham [2] and In Hong Chang [1,*]

[1] Department of Computer Science and Statistics, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju 61452, Korea; imk92315@chosun.kr (Y.S.K.); csssig84@gmail.com (K.Y.S.)
[2] Department of Industrial and Systems Engineering, Rutgers University, 96 Frelinghuysen Road, Piscataway, NJ 08855-8018, USA; hopham@rci.rutgers.edu
* Correspondence: ihchang@chosun.ac.kr; Tel.: +82-62-230-6621

Abstract: In the past, because computer programs were restricted to perform only simple functions, the dependence on software was not large, resulting in relatively small losses after a failure. However, with the development of the software market, the dependence on software has increased considerably, and software failures can cause significant social and economic losses. Software reliability studies were previously conducted under the assumption that software failures occur independently. However, as software systems become more complex and extremely large, software failures are becoming frequently interdependent. Therefore, in this study, a software reliability model is developed under the assumption that software failures occur in a dependent manner. We derive the software reliability model through the number of software failure and fault detection rate assuming point symmetry. The proposed model proves good performance compared with 21 previously developed software reliability models using three datasets and 11 criteria. In addition, to find the optimal release time, a cost model using the developed software reliability model was presented. To determine this release time, four parameters constituting the software reliability model were changed by 10%. By comparing the change in the cost model and the optimal release time, it was found that parameter b had the greatest influence.

Keywords: non-homogeneous Poisson process; dependence failure; software reliability; software reliability model; cost model

Citation: Kim, Y.S.; Song, K.Y.; Pham, H.; Chang, I.H. A Software Reliability Model with Dependent Failure and Optimal Release Time. *Symmetry* **2022**, *14*, 343. https://doi.org/10.3390/sym14020343

Academic Editor: Ming-Chin Chuang

Received: 31 December 2021
Accepted: 7 February 2022
Published: 8 February 2022

1. Introduction

Software, one of the main components of a computer, plays an important role in the operation of physical devices. Software was originally developed with the ability to perform extremely small or simple functions. Currently, however, embedded systems that perform multiple functions are being developed. With the rapid development of the software market, technology has also developed, and software is now being used in all fields. Recently, the Internet of Things (IoT) based on the combination of various software, has been commercialized. Furthermore, AIoT (Artificial Intelligence of Things) combined AI (Artificial Intelligence) with IoT (Intelligence of Things) is developing [1]. It means that software has become a very important part not only in the industrial field but also in our daily life.

A software failure is caused by various faults (coding or system errors, etc.). In the past, software failures caused relatively small losses because the degree of software dependence was not as large. However, in today's world, the degree of dependence on software is extremely high, and thus software failures can cause significant social and economic losses. Therefore, we measured the software reliability, which indicates the ability of a software program to avoid failure for a set period of time and refers to how long the software can be used without such a failure.

Early research on software reliability was conducted based on the assumption that software failures occur independently. Goel and Okumoto proposed the GO model, which is the most basic non-homogeneous Poisson process (NHPP) software reliability growth model [2]. The Hossain, Dahiya, Goel and Okumoto (HDGO) model further extended the GO model [3]. Yamada et al., Ohba, and Zhang et al. [4–6] proposed an NHPP S-shaped curve model in which the cumulative number of software failures increases to the S curve. In addition, Yamada et al. [7] proposed a new model in which the test effort invested during phase was reflected in the software reliability model. It is a model that reflects even the resources consumed for testing in the previously developed model. Furthermore, Yamada et al. [8] developed a software reliability model with a constant fault detection rate of $b(t) = b$, assuming incomplete debugging, in which faults detected during the test phase were corrected and removed.

The model developed by extending the above approach involved a generalized incomplete debugging-error detection rate model. Here, the fault detection rate $b(t)$ of the model is not a constant but rather a different function [9–14]. It started from the software error causing the failure being immediately eliminated and so a new error can be generated [9]. It progressed to that during the fault removal process, whether the fault is removed successfully or not, new faults are generated with a constant probability [13,14]. In addition, because the operating environment of the software is operated differently for each software program, a comparison is difficult to achieve. Therefore, in [15–17], a software reliability model was developed considering uncertain factors in the operating environment. Currently, research using non-parametric methods such as deep learning or machine learning is also being conducted [18–21].

Recently, finding the most optimal model for reliability prediction is an important concern. Through combination of analytic hierarchy method (AHP), hesitant fuzzy (HF) sets and techniques for order of preference by similarity to ideal solution (TOPSIS), Sahu et al., Ogundoyin et al., and Rafi et al. [22–24] found the most optimal software reliability model.

However, software failures often occur in a dependent manner because the developed software is composed of extremely complex structures [25]. Here, the dependent failure means that one failure affects other failures or increases the failure probability of other equipment [26,27]. There are two main types of dependent failure. A common cause failure is when several pieces of software fail simultaneously due to a common cause, and a cascading failure is a case in which a part of the system fails and affects other software as well. A software reliability model assuming a dependent failure was developed from the number of software failures and the fault detection rate, which have a dependency relationship in a software reliability model assuming incomplete debugging [28]. In addition, Lee et al. [29] presented a model that assumes that if past software failures are not corrected well, they will continue to have an effect.

In this study, a new software reliability model is developed under the assumption that software failures occur in a dependent manner. It is suitable for a general environment. We show the superiority of the newly developed dependent software reliability model through a comparison under various criteria. In addition, determining the optimal release time of the developed software is also important. If the test period is long, the software will be reliable, but the software development cost will increase. If the test period is short, the reliability of the product may decrease. Therefore, it is important to find a balance between time to market and minimum cost taking into account the installation costs, test costs, and error removal costs, etc. We propose a cost model that combines the proposal software reliability model and the cost coefficient [30–33]. In addition, among the various parameters of the proposed model, we propose a parameter that has a significant influence on predicting the number of cumulative failures through a variation in the cost model for changes in the parameters [34,35]. Section 2 introduces a new dependent software reliability model and its mathematical derivation. Section 3 introduces the data and criteria, as well as numerical results. Section 4 describes the optimal release time, and finally, in Section 5, we present our conclusions

2. New Dependent Software Reliability Model

Software reliability refers to the probability that the software will not fail a system for a certain period of time under certain conditions. In other words, it evaluates "how long the software can be used without failure". The reliability function used to evaluate this is as follows:

$$R(t) = P(T > t) = \int_t^\infty f(u)du \tag{1}$$

This denotes the probability of the software operating without failure over a specific time t. Here, the probability density function $f(t)$ assumes the software failure time or lifetime as a random variable T. When measuring reliability function $R(t)$, it is assumed that it follows an exponential distribution with parameter λ. In addition, it is assumed that the number of failures occurring in given unit time is a Poisson distribution with parameter λ. When λ is a constant, it is the most basic form, and is called a homogeneous Poisson process. Extending this process, many researchers adapt a model where λ is an intensity function $\lambda(t)$ that changes with time rather than a single constant by setting λ as a non-homogeneous Poisson process(NHPP) rather than as a homogeneous Poisson process.

$$\Pr\{N(t) = n\} = \frac{\{m(t)\}^n}{n!} e^{-m(t)}, n = 0, 1, 2, \cdots, \ t \geq 0 \tag{2}$$

In Equation (2), $N(t)$ is the Poisson probability density function with the time dependent parameter $m(t)$. The $m(t)$ is a mean value function which is the integral of $\lambda(t)$ from 0 to t in Equation (3). The $\lambda(t)$ is the intensity function indicating the number of instantaneous failures at time t.

$$m(t) = \mathrm{E}[N(t)] = \int_0^t \lambda(s)ds \tag{3}$$

A general class of NHPP software reliability models was proposed by Equation (9) to summarize the existing NHPP models as follows:

$$\frac{dm(t)}{dt} = b(t)[a(t) - m(t)] \tag{4}$$

where, the $m(t)$ is calculated using the relationship between the number of failures $a(t)$ at each time point and a fault detection rate $b(t)$ assuming point symmetry in Equation (4). Various software reliability models have been developed based on the assumption that software failures occur independently.

However, software failures occur not only independently but also dependently. If the failure is not completely fixed, it will continue to affect the next failure. In addition, as the system becomes more complex, the relationship between failure and failure also shows the dependent relationship because of the dependent combination of several software. Therefore, in this study, we assume that failure is dependent on other failures. The mean value function $m(t)$ based on NHPP software reliability model using the differential equation is as follows:

$$\frac{dm(t)}{dt} = b(t)[a(t) - m(t)]m(t) \tag{5}$$

In Equation (5), the $m(t)$ is multiplied once more to assume that the failure occurring from 0 to t affects another failure. We assume:

$$a(t) = a(1 + \alpha t), \ b(t) = \frac{b}{1 + ce^{-bt}} \tag{6}$$

where, $a(t)$ is the number of software failures at each time point and $b(t)$ is the fault detection rate. Parameter a is the expected number of faults, α is the increasing rate of the number of faults, b is the shape parameter, and c is the scale parameter. When time t changes, the change according to the values of parameters b and c in the fault detection rate $b(t)$ are as shown in Figure 1. When b is 1, it is blue, and when it is 1.5, it is red. In addition, when c is 1, it is a dashed line, and when it is 2, it is a dotted line. It can be seen that the larger b is, the larger the $b(t)$ is.

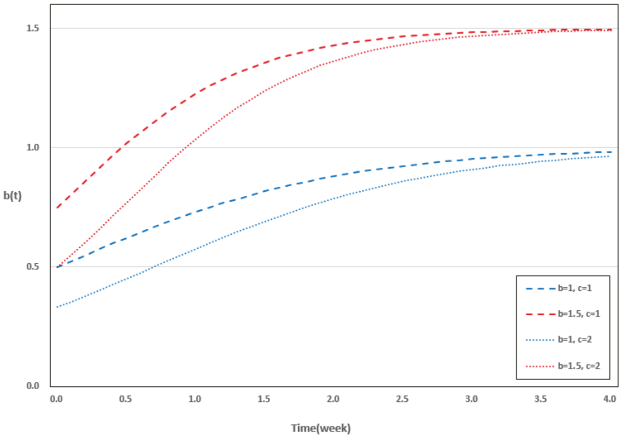


Figure 1. $b(t)$ according to the changes in parameters b and c.

When solving the differential equation by substituting $a(t)$ and $b(t)$ for $m(t)$ in Equations (5) and (6), we obtain Equation (7):

$$m(t) = \frac{\left(c+e^{bt}\right)^{a}\left(\frac{c+e^{bt}}{c}\right)^{a\alpha t} e^{\frac{a\alpha Li_2\left(\frac{e^{bt}+c}{c}\right)}{b}}}{\int \frac{\left(c+e^{bt}\right)^{a}\left(\frac{c+e^{bt}}{c}\right)^{a\alpha t} e^{\frac{a\alpha Li_2\left(\frac{e^{bt}+c}{c}\right)}{b}} be^{bt}}{c+e^{bt}} dt + C} \tag{7}$$

where $Li_s(x) = \sum_{n=1}^{\infty} \frac{x^n}{n^s}$ is a polylogarithm when $s = 2$. At this time, $\alpha = 0$ in $a(t)$.

$$m(t) = \frac{h\left(c+e^{bt}\right)^{a}}{h\left[\int_0^t \frac{\left(e^{bx}+c\right)^{a} be^{bx}}{c+e^{bx}} dx\right] + (1+c)^{a}} \tag{8}$$

where h is the number of initial failures. In Equation (8), $\int_0^t \frac{\left(e^{bx}+c\right)^{a} be^{bx}}{c+e^{bt}} dx$ is calculated through an integration using substitution. When $u = c + be^{bx}$ and $du = be^{bx}dx$, it is the same as in Equation (9).

$$\int_0^t \frac{\left(e^{bx}+c\right)^{a} be^{bx}}{c+e^{bt}} dx = \int_{c+b}^{c+be^{bt}} \frac{u^a}{u} du = \int_{c+b}^{c+be^{bt}} u^{a-1} du = \frac{u^a}{a} = \frac{\left(c+e^{bt}\right)^{a}}{a} \tag{9}$$

Substituting the result of the substitution integration into Equation (8), the final $m(t)$ is given by Equation (10).

$$m(t) = \frac{a}{1 + \frac{a}{h}\left(\frac{1+c}{c+e^{bt}}\right)^{a}} \tag{10}$$

This can be presented as a general model of a dependent failure occurrence in the software reliability model. When $t = 0$, $m(t)$ is $m(0) = ah/(a+h)$. Table 1 shows the value of $m(t)$ of the existing software reliability model and the model proposed in this study. From models 19–22, it is assumed that a failure occurs in a dependent manner.

Table 1. Software reliability models.

No.	Model	Mean Value Function	Note
1	Goel-Okumoto (GO) [2]	$m(t) = a\left(1 - e^{-bt}\right)$	Concave
2	Hossain-Dahiya (HDGO) [3]	$m(t) = \log\left[\frac{(e^{a} - c)}{\left(e^{ae^{-bt}} - c\right)}\right]$	Concave
3	Yamada et al. (DS) [4]	$m(t) = a\left(1 - (1 + \mathrm{bt})e^{-bt}\right)$	S-Shape
4	Ohba (IS) [5]	$m(t) = \frac{a\left(t1 - e^{-bt}\right)}{1 + \beta e^{-bt}}$	S-Shape
5	Zhang et al. (ZFR) [6]	$m(t) = \frac{a}{p - \beta}\left[1 - \left(\frac{(1+\alpha)e^{-bt}}{1 + \alpha e^{-bt}}\right)^{\frac{c}{b}(p - \beta)}\right]$	S-Shape
6	Yamada et al. (YE) [7]	$m(t) = a\left(1 - e^{-\gamma\alpha(1 - e^{-\beta t})}\right)$	Concave
7	Yamada et al. (YR) [7]	$m(t) = a\left(1 - e^{-\gamma\alpha(1 - e^{-\beta t^{2}/2})}\right)$	S-Shape
8	Yamada et al. (YID 1) [8]	$m(t) = \frac{ab}{\alpha + b}\left(e^{\alpha t} - e^{-bt}\right)$	Concave
9	Yamada et al. (YID 2) [8]	$m(t) = a\left(1 - e^{-bt}\right)\left(1 - \frac{\alpha}{b}\right) + \alpha at$	Concave
10	Pham-Zhang (PZ) [9]	$m(t) = \frac{\left((c + a)\left[1 - e^{-bt}\right] - \left[\frac{ab}{b - \alpha}\right]\left(e^{-at} - e^{-bt}\right)\right)}{1 + \beta e^{-bt}}$	Both
11	Pham et al. (PNZ) [10]	$m(t) = \frac{a\left(1 - e^{-bt}\right)\left(1 - \frac{\alpha}{b}\right) + \alpha at}{1 + \beta e^{-bt}}$	Both
12	Teng-Pham (TP) [11]	$m(t) = \frac{a}{p - q}\left[1 - \left(\frac{\beta}{\beta + (p - q)\ln\left(\frac{c + e^{bt}}{c + 1}\right)}\right)^{\alpha}\right]$	S-Shape
13	Kapur et al. (KSRGM) [12]	$m(t) = \frac{A}{1 - \alpha}\left[1 - ((\left(1 + bt + \frac{b^{2}t^{2}}{2}\right)e^{-bt})^{p(1 - \alpha)}\right]$	S-Shape
14	Roy et al. (RMD) [13]	$m(t) = a\alpha\left[1 - e^{-bt}\right] - \left[\frac{ab}{b - \beta}\left(e^{-\beta t} - e^{-bt}\right)\right]$	Concave
15	Pham (IFD) [14]	$m(t) = a\left(1 - e^{-bt}\right)(1 + (b + d)t + bdt^{2})$	Concave
16	Pham (Vtub) [15]	$m(t) = \mathrm{N}\left[1 - \left(\frac{\beta}{\beta + a^{bt} - 1}\right)^{\alpha}\right]$	S-Shape
17	Chang et al. (TC) [16]	$m(t) = \mathrm{N}\left[1 - \left(\frac{\beta}{\beta + (at)^{b}}\right)^{\alpha}\right]$	Both
18	Song et al. (3P) [17]	$m(t) = \mathrm{N}\left[1 - \left(\frac{\beta}{\beta - \frac{a}{b}\ln\left(\frac{(1+c)e^{-bt}}{1 + ce^{-bt}}\right)}\right)\right]$	S-Shape
19	Pham (DP1) [28]	$m(t) = \alpha(1 + \beta t)\left(\beta t + \left(e^{-\beta t} - 1\right)\right)$	Concave, Dependent
20	Pham (DP2) [28]	$m(t) = m_{0}\left(\frac{\gamma t + 1}{\gamma t_{0} + 1}\right)e^{-\gamma(t - t_{0})} +$ $\alpha(\gamma t + 1)\left(\gamma t - 1 + (1 - \gamma t_{0})e^{-\gamma(t - t_{0})}\right)$	Concave, Dependent
21	Lee et al. (DPF) [29]	$m(t) = \frac{a}{1 + \frac{a}{h}\left(\frac{b + c}{c + be^{bt}}\right)^{\frac{a}{b}}}$	S-Shape, Dependent
22	Proposed Model	$m(t) = \frac{a}{1 + \frac{a}{h}\left(\frac{1 + c}{c + e^{bt}}\right)^{a}}$	S-Shape, Dependent

3. Numerical Examples

3.1. Data Information

Datasets 1 and 2 are derived from the online communication system (OCS) of ABC Software Co. and uses data accumulated over a 12-week period. Datasets 1 and 2 show that the cumulative number of failures at $t = 1, 2, \cdots, 12$ is $14, 17, \cdots, 81$, and $11, 17, \cdots, 81$, respectively [14]. Dataset 3 is the test data of a medical record system consisting of 188 software titles and data for one of three releases. It shows that the cumulative number of failures is $90, 107, \cdots, 204$ for $t = 1, 2, \cdots, 17$, respectively [36]. Table 2 shows the accumulated failure data for datasets 1, 2, and 3. We compare the fit between the software reliability models with two failure datasets obtained from OCS and one dataset from Lee et al. [29], which showed good performance as the dependent models (DPF).

Table 2. Cumulative number of software failure datasets.

Index	Dataset 1		Dataset 2		Dataset 3	
	Failures	Cumulative Failures	Failures	Cumulative Failures	Failures	Cumulative Failures
1	14	14	11	11	90	90
2	3	17	6	17	17	107
3	4	21	0	17	19	126
4	7	28	5	22	19	145
5	7	35	5	27	26	171
6	18	53	25	52	17	188
7	8	61	10	62	1	189
8	4	65	6	68	1	190
9	2	67	2	70	0	190
10	9	76	10	80	0	190
11	1	77	0	80	2	192
12	4	81	1	81	0	192
13					0	192
14					0	192
15					11	203
16					0	203
17					1	204

3.2. Criteria

This study compares various independent and dependent software reliability models and the proposed model introduced Table 1 using 11 criteria. Based on the difference between the actual observed value and the estimated value, we would like to find a better model by comparing it with criteria reflecting the number of parameters used in each model.

First, the mean squared error (MSE) is defined as the sum of squares of the distance between the estimated value and the actual value when considering the number of parameters and the number of observations [37].

$$\text{MSE} = \frac{\sum_{i=1}^{n}\left(\hat{m}(t_i) - y_i\right)^2}{n-m} \tag{11}$$

where $\hat{m}(t_i)$ is the estimated value of the model $m(t)$, y_i is the actual observed value, n is the number of observations, and m is the number of parameters in each model.

Second, the mean absolute error (MAE) defines the difference between the estimated number of failures and the actual value considering the number of parameters and the number of observations as the sum of the absolute values [38].

$$\text{MAE} = \frac{\sum_{i=1}^{n}\left|\hat{m}(t_i) - y_i\right|}{n-m} \tag{12}$$

Third, Adj_R^2 is the modified coefficient of determination of the regression equation and determines how much explanatory power it has in consideration of the number of parameters [39].

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{n}\left(\hat{m}(t_i) - y_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y_i}\right)^2},\ \text{Adj\_R}^2 = 1 - \frac{\left(1 - R^2\right)(n-1)}{n-m-1} \tag{13}$$

Fourth, the predictive ratio risk (PRR) is obtained by dividing the distance from the actual value to the estimated value by the estimated value in relation to the model estimation [40].

$$\text{PRR} = \sum_{i=1}^{n}\left(\frac{\hat{m}(t_i) - y_i}{\hat{m}(t_i)}\right)^2 \tag{14}$$

Fifth, the predictive power (PP) is obtained by dividing the distance from the actual value to the estimated value by the actual value [41].

$$\mathrm{PP} = \sum_{i=1}^{n} \left(\frac{\widehat{m}(t_i) - y_i}{y_i} \right)^2 \tag{15}$$

Sixth, Akaike's information criterion (AIC) was used to compare likelihood function maximization. This is applied to maximize the Kullback–Leibler level between the probability distribution of the model and the data [42].

$$\begin{gathered}\mathrm{AIC} = -2\log L + 2m \\ L = \prod_{i=1}^{n} \frac{(m(t_i)-m(t_{i-1}))^{y_i-y_{i-1}}}{(y_i-y_{i-1})!} e^{-(m(t_i)-m(t_{i-1}))} \\ logL = \sum_{i=1}^{n} \{(y_i - y_{i-1}) \ln(m(t_i) - m(t_{i-1})) - (m(t_i) - m(t_{i-1})) \\ - \ln((y_i - y_{i-1})!)\}\end{gathered} \tag{16}$$

Seventh, the predicted relative variation (PRV) is the standard deviation of the prediction bias and is defined as [43]

$$\mathrm{PRV} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \widehat{m}(t_i) - Bias)^2}{n-1}} \tag{17}$$

Here, the bias is $\sum_{i=1}^{n} \left[\frac{\widehat{m}(t_i) - y_i}{n} \right]$.

The root mean square prediction error (RMSPE) can estimate the closeness with which the model predicts the observation [44]:

$$\mathrm{RMSPE} = \sqrt{Variance^2 + Bias^2} \tag{18}$$

Ninth, the mean error of prediction (MEOP) sums the absolute value of the deviation between the actual data and the estimated curve and is defined as [38]

$$\mathrm{MEOP} = \frac{\sum_{i=1}^{n} |\widehat{m}(t_i) - y_i|}{n - m + 1} \tag{19}$$

Tenth, the Theil statistic (TS) is the average percentage of deviation over all periods with regard to the actual values. The closer the Theil statistic is to zero, the better the prediction capability of the model. This is defined as [45]

$$\mathrm{TS} = 100 * \sqrt{\frac{\sum_{i=1}^{n} (y_i - \widehat{m}(t_i))^2}{\sum_{i=1}^{n} {y_i}^2}} \% \tag{20}$$

Eleventh, it takes into account the tradeoff between the uncertainty in the model and the number of parameters in the model by slightly increasing the penalty each time parameters are added to the model when the sample is considerably small [46].

$$\mathrm{PC} = \left(\frac{n-m}{2} \right) \log \left(\frac{\sum_{i=1}^{n} (\widehat{m}(t_i) - y_i)^2}{n} \right) + m \left(\frac{n-1}{n-m} \right) \tag{21}$$

Based on the above criteria, we compared the proposed model with the existing NHPP software reliability model. When Adj_R^2 is closer to 1, and the other 10 criteria are closer to 0, it indicates a better fit. Using R and MATLAB, the parameters of each model were estimated through the LSE method, and the goodness of fit is calculated to compare the superiority. This is a method of estimating parameters through the difference between the model in Table 1 and the actual number of failures in Table 2, and follows LSE $= \sum_{t=1}^{n}(y_t - m(t))^2$ [47].

3.3. Results of Dataset 1

Table 3 shows the estimated values for the parameters of each model obtained using dataset 1. Each parameter of the proposed model is represented by $\hat{a} = 80.0907$, $\hat{b} = 0.07231$, $\hat{c} = 15.9288$, and $\hat{h} = 9.8182$. Figure 2 shows the result of calculating the estimated value of $m(t)$ at each time point based on the cumulative number of failures at each time point in dataset 1 and each model equation. The black dotted line represents the actual data, and the dark red solid line represents the predicted failure value at each time point of the proposed model. Compared with other models, it shows the predicted value closest to the actual value.

Table 3. Parameter estimation of model from dataset 1.

No.	Model	Estimation
1	GO	$\hat{a} = 191.3881$, $\hat{b} = 0.0483$
2	HDOG	$\hat{a} = 191.3880$, $\hat{b} = 0.04832$, $\hat{c} = 1.3929$
3	DS	$\hat{a} = 92.0916$, $\hat{b} = 0.3034$
4	IS	$\hat{a} = 88.9815$, $\hat{b} = 0.3274$, $\hat{\beta} = 3.9383$
5	ZFR	$\hat{a} = 14.6285$, $\hat{b} = 0.21179$, $\hat{\alpha} = 33.5808$ $\hat{\beta} = 0.0304$, $\hat{c} = 15.1085$, $\hat{p} = 0.2085$
6	YE	$\hat{a} = 212.1517$, $\hat{\alpha} = 0.2021$ $\hat{\beta} = 0.00568$, $\hat{\gamma} = 38.0032$
7	YR	$\hat{a} = 101.8036$, $\hat{\alpha} = 0.5271$ $\hat{\beta} = 0.0276$, $\hat{\gamma} = 3.3321$
8	YID1	$\hat{a} = 181.0676$, $\hat{b} = 0.05131$, $\hat{\alpha} = 0.00114$
9	YID2	$\hat{a} = 140.9842$, $\hat{b} = 0.06636$, $\hat{\alpha} = 0.0126$
10	PZ	$\hat{a} = 27.3845$, $\hat{b} = 0.41396$, $\hat{\alpha} = 0.1349$ $\hat{\beta} = 4.5437$, $\hat{c} = 63.7781$
11	PNZ	$\hat{a} = 27.3845$, $\hat{b} = 0.0218$ $\hat{\alpha} = 0.0000332$, $\hat{\beta} = 0.0000684$
12	TP	$\hat{a} = 0.8506$, $\hat{b} = 0.38022$, $\hat{\alpha} = 0.6721$, $\hat{\beta} = 0.000333$ $\hat{c} = 114.6551$, $\hat{p} = 0.0122$, $\hat{q} = 0.00326$
13	KSRGM	$\hat{A} = 3.2070$, $\hat{b} = 8.76921$ $\hat{\alpha} = 0.9764$, $\hat{p} = 0.4157$
14	RMD	$\hat{a} = 78.5350$, $\hat{b} = 0.21915$ $\hat{\alpha} = 1.3358$, $\hat{\beta} = 0.2192$
15	IFD	$\hat{a} = 7.6597$, $\hat{b} = 0.84452$, $\hat{d} = 0.00171$
16	Vtub	$\hat{a} = 1.8954$, $\hat{b} = 0.70887$, $\hat{\alpha} = 6.7593$ $\hat{\beta} = 62.2968$, $\hat{N} = 83.1687$
17	TC	$\hat{a} = 0.1736$, $\hat{b} = 1.33331$, $\hat{\alpha} = 11.5457$ $\hat{\beta} = 18.8347$, $\hat{N} = 105.1851$
18	3P	$\hat{a} = 1.4640$, $\hat{b} = 0.3299$, $\hat{\beta} = 0.6008$ $\hat{N} = 94.1037$, $\hat{c} = 37.8421$
19	DP1	$\hat{\alpha} = 0.1104$, $\hat{\beta} = 2.5829$
20	DP2	$\hat{\alpha} = 46197.046$, $\hat{\gamma} = 0.00451$ $\widehat{t_0} = 3.7402$, $\widehat{m_0} = 30.01197$
21	DPF	$\hat{a} = 80.3065$, $\hat{b} = 0.06122$ $\hat{c} = 13.9314$, $\hat{h} = 9.8182$
22	Proposed model	$\hat{a} = 80.0907$, $\hat{b} = 0.07231$ $\hat{c} = 15.9288$, $\hat{h} = 9.8182$

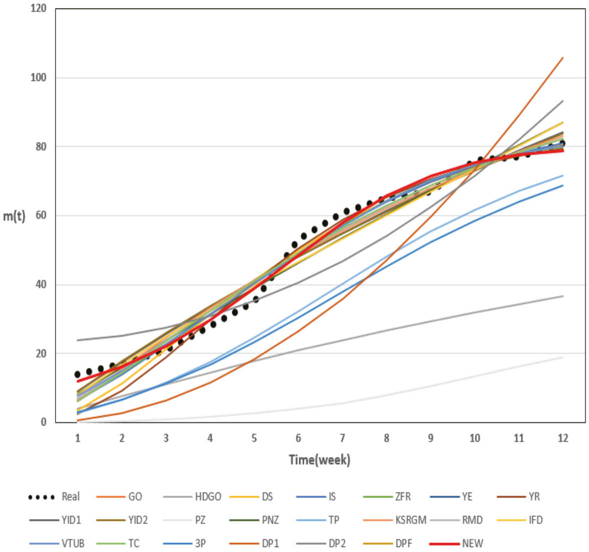


Figure 2. Prediction of all models for dataset 1.

Table 4 shows the results of calculating the criteria of each model using the parameters obtained through dataset 1. As a result, the values of MSE, MAE, PRR, PP, PRV, RMSPE, MEOP, TS, PC of the proposed model show the smallest values of 9.9274, 3.2140, 0.0647, 0.0577, 2.6866, 2.6870, 2.8569, 4.6682, and 13.0594, respectively. The AIC shows the second smallest value at 73.4605. In addition, Adj_R^2 is 0.9821, which is the closest to 1. The DPF model shows the highest value with AIC = 73.2850, and the second highest result for the other criteria. The model with the third highest criterion is the Vtub model.

Table 4. Comparison of all criteria from dataset 1.

No.	Model	MSE	MAE	Adj_R2	PRR	PP	AIC	PRV	RMSPE	MEOP	TS	PC
1	GO	21.1918	4.4966	0.9627	0.4187	0.2758	80.5308	4.3889	4.3892	4.0878	7.6254	16.5565
2	HDOG	23.5465	4.9962	0.9581	0.4187	0.2758	82.5308	4.3889	4.3892	4.4966	7.6254	16.5875
3	DS	22.4994	3.8718	0.9604	9.4838	0.7292	92.7861	4.4119	4.5135	3.5198	7.8572	16.8558
4	IS	16.8528	3.8882	0.9700	1.4452	0.3769	80.9153	3.6786	3.7104	3.4994	6.4512	15.0824
5	ZFR	24.2415	5.8035	0.9540	1.2042	0.3441	86.8823	3.6068	3.6338	4.9744	6.3174	18.4848
6	YE	26.5773	5.6309	0.9519	0.4182	0.2753	84.5796	4.3963	4.3965	5.0052	7.6380	16.9984
7	YR	33.2210	5.1600	0.9398	23.6437	0.9460	103.5558	4.6865	4.8967	4.5867	8.5395	17.8909
8	YID1	23.6037	4.9679	0.9579	0.4140	0.2764	82.4537	4.3944	4.3946	4.4711	7.6347	16.5984
9	YID2	23.8374	5.0079	0.9575	0.4070	0.2771	82.5743	4.4162	4.4163	4.5071	7.6724	16.6427
10	PZ	217.3085	17.6277	0.5983	0.5816	1.2389	84.1256	4.7893	11.3435	15.4243	20.4300	24.8053
11	PNZ	4000.777	68.0533	−6.2441	2.6223	10.4504	142.6741	25.7721	52.1779	60.4918	93.7126	37.0551
12	TP	31.0301	7.0496	0.9387	1.6004	0.3946	89.3412	3.7102	3.7518	5.8747	6.5247	21.7987
13	KSRGM	26.0975	5.4865	0.9527	1.5345	0.4198	88.8357	4.3440	4.3556	4.8769	7.5688	16.9255
14	RMD	21.9129	5.0225	0.9604	1.1967	0.3659	84.6918	3.9784	3.9909	4.4644	6.9355	16.2264
15	IFD	29.9616	5.5723	0.9467	0.653	0.3059	87.6572	4.9344	4.9498	5.0151	8.6017	17.6717
16	Vtub	18.9012	4.8111	0.9650	0.7591	0.2784	82.2273	3.4511	3.4667	4.2097	6.0252	16.2579
17	TC	26.6474	5.8343	0.9507	1.8283	0.4312	89.1447	4.0938	4.1159	5.1050	7.1541	17.4601
18	3P	21.7357	5.0238	0.9599	1.4395	0.3767	84.9059	3.6859	3.7164	4.3958	6.4613	16.7470
19	DP1	361.825	19.1467	0.3631	448.095	3.2745	174.8811	14.9748	17.8944	17.4061	31.5087	30.7442
20	DP2	113.5394	11.4623	0.7945	0.5996	1.0133	109.7792	9.0867	9.0870	10.1888	15.7870	22.8067
21	DPF	9.9490	3.2278	0.9819	0.0689	0.0606	73.2850	2.6894	2.6899	2.8692	4.6732	13.0680
22	Proposed model	9.9274	3.2140	0.9821	0.0647	0.0577	73.4605	2.6866	2.6870	2.8569	4.6682	13.0594

3.4. Results of Dataset 2

Table 5 shows the estimated values for the parameters of each model obtained using dataset 2. Each parameter of the proposed model is represented as $\hat{a} = 79.1444$, $\hat{b} = 0.2001$, $\hat{c} = 72.3208$, and $\hat{h} = 9.3327$. Figure 3 shows the results of calculating the estimated value of $m(t)$ for each point in time based on the cumulative number of failures at each time point in dataset 2 and each model equation. Here, the black dotted line represents the actual data, whereas the dark red solid line represents the predicted failure value at each time point of the proposed model. Compared with the other models, the predicted value is closest to the actual value.

Table 6 shows the results of calculating the criteria of each model using the parameters obtained through dataset 2. As a result, the values of MSE, MAE, PRR, PP, AIC, PRV, RMSPE, MEOP, TS, and PC of the proposed model are 18.9722, 4.3544, 0.1615, 0.1482, 92.2155, 3.7139, 3.7145, 3.8706, 6.3751, and 15.6500, respectively, which show the smallest criteria. Adj_R^2 is 0.9723, which is the closest to 1. The model with the second highest criterion is DPF, and Vtub is the third best-fitting model.

Table 5. Parameter estimation of model from dataset 2.

No.	Model	Estimation
1	GO	$\hat{a} = 518.0607, \hat{b} = 0.0156$
2	HDOG	$\hat{a} = 518.0607, \hat{b} = 0.01561, \hat{c} = 15.4479$
3	DS	$\hat{a} = 105.5701, \hat{b} = 0.2548$
4	IS	$\hat{a} = 85.7894, \hat{b} = 0.4918, \hat{\beta} = 13.4498$
5	ZFR	$\hat{a} = 4.3416, \hat{b} = 0.3506, \hat{\alpha} = 67.6487$ $\hat{\beta} = 0.00128, \hat{c} = 57.1077, \hat{p} = 0.0548$
6	YE	$\hat{a} = 583.6809, \hat{\alpha} = 0.0700$ $\hat{\beta} = 0.00224, \hat{\gamma} = 88.6105$
7	YR	$\hat{a} = 0.8619, \hat{\alpha} = 5.0727$ $\hat{\beta} = 0.0000000232, \hat{\gamma} = 0.3849$
8	YID1	$\hat{a} = 369.1037, \hat{b} = 0.02206, \hat{\alpha} = 0.00446$
9	YID2	$\hat{a} = 174.7977, \hat{b} = 0.04618, \hat{\alpha} = 0.0292$
10	PZ	$\hat{a} = 80.3516, \hat{b} = 0.5195, \hat{\alpha} = 4.0448$ $\hat{\beta} = 15.4266, \hat{c} = 4.4447$
11	PNZ	$\hat{a} = 82.6869, \hat{b} = 0.5259$ $\hat{\alpha} = 0.0013, \hat{\beta} = 15.5683$
12	TP	$\hat{a} = 0.2772, \hat{b} = 0.5328, \hat{\alpha} = 0.5551, \hat{\beta} = 0.000359$ $\hat{c} = 76.4359, \hat{p} = 0.5610, \hat{q} = 0.5583$
13	KSRGM	$\hat{A} = 85.4315, \hat{b} = 0.3829$ $\hat{\alpha} = 0.0353, \hat{p} = 1.5904$
14	RMD	$\hat{a} = 102.8360, \hat{b} = 0.2355$ $\hat{\alpha} = 1.0649, \hat{\beta} = 0.2355$
15	IFD	$\hat{a} = 21.4202, \hat{b} = 0.2758, \hat{d} = 0.00001004$
16	Vtub	$\hat{a} = 2.1725, \hat{b} = 0.7383, \hat{\alpha} = 48.1784$ $\hat{\beta} = 961.5799, \hat{N} = 81.1633$
17	TC	$\hat{a} = 0.1524, \hat{b} = 1.9761, \hat{\alpha} = 21.6626$ $\hat{\beta} = 22.4388, \hat{N} = 87.7121$
18	3P	$\hat{a} = 2.3685, \hat{b} = 0.4823, \hat{\beta} = 1.1973$ $\hat{N} = 94.4688, \hat{c} = 59.8790$
19	DP1	$\hat{\alpha} = 0.01104, \hat{\beta} = 8.2449$
20	DP2	$\hat{\alpha} = 4.1249, \hat{\gamma} = 0.0000000161$ $\widehat{t_0} = 0.0000168, \widehat{m_0} = 0.000213$
21	DPF	$\hat{a} = 79.1447, \hat{b} = 0.1928$ $\hat{c} = 69.3637, \hat{h} = 9.2699$
22	Proposed model	$\hat{a} = 79.1444, \hat{b} = 0.2001$ $\hat{c} = 72.3208, \hat{h} = 9.3327$

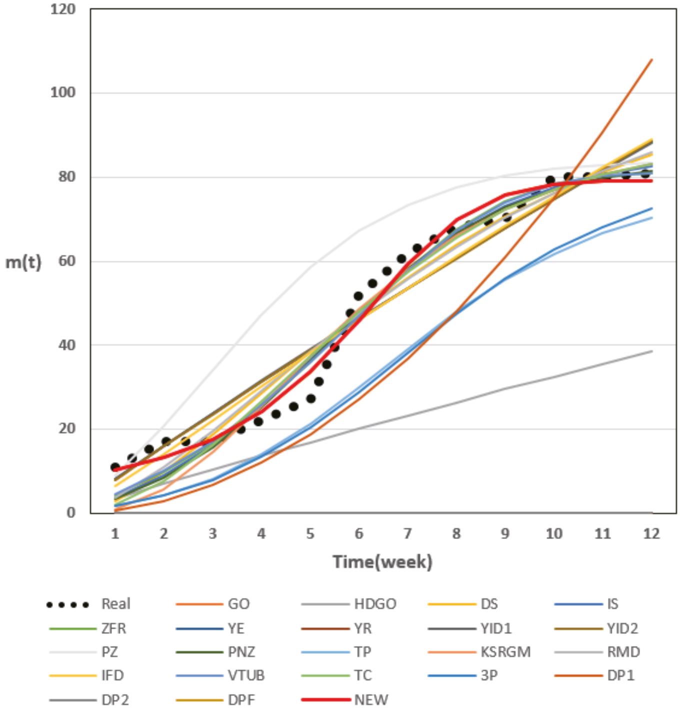


Figure 3. Prediction of all models for dataset 2.

Table 6. Comparison of all criteria from dataset 2.

No.	Model	MSE	MAE	Adj_R^2	PRR	PP	AIC	PRV	RMSPE	MEOP	TS	PC
1	GO	52.7922	6.9705	0.9252	0.4713	0.6631	113.9369	6.9157	6.9267	6.3369	11.8896	21.1202
2	HDOG	58.6580	7.7450	0.9159	0.4713	0.6631	115.9369	6.9157	6.9267	6.9705	11.8896	20.6949
3	DS	40.0324	5.9373	0.9433	8.5312	1.0273	116.8063	5.9998	6.0299	5.3975	10.3536	19.7368
4	IS	30.7961	5.1688	0.9559	5.1137	0.8444	102.4472	4.9413	5.0132	4.6519	8.6150	17.7954
5	ZFR	42.3215	7.0918	0.9353	3.2756	0.7195	104.7204	4.7496	4.8001	6.0787	8.2459	20.1564
6	YE	66.0197	8.6990	0.9038	0.4695	0.6677	117.922	6.9138	6.9280	7.7325	11.8923	20.6380
7	YR	4668.125	73.375	−5.7994	4.00×10^{17}	12.000	2807.273	28.0112	56.369	65.2222	100.000	37.6722
8	YID1	58.7470	7.7242	0.91571	0.4677	0.6707	115.8113	6.9207	6.9319	6.9518	11.8987	20.7017
9	YID2	58.9848	7.7955	0.91544	0.4730	0.6551	116.2140	6.9390	6.9463	7.0159	11.9227	20.7199
10	PZ	42.3248	6.7234	0.9371	11.0678	1.0170	110.5663	5.0554	5.1787	5.8830	8.9070	19.0795
11	PNZ	35.2459	5.7358	0.9486	6.5289	0.9046	104.9515	4.8955	5.0492	5.0985	8.6893	18.1275
12	TP	63.8102	10.1380	0.8983	4.8099	0.8346	112.9315	5.0230	5.3563	8.4484	9.2430	23.6011
13	KSRGM	50.5240	6.8779	0.9265	114.740	1.4819	135.0029	5.8715	6.0461	6.1137	10.4035	19.5679
14	RMD	49.4957	7.4822	0.9279	3.9839	0.8882	116.9778	5.9847	5.9985	6.6509	10.297	19.4857
15	IFD	54.1681	7.6923	0.9223	0.7680	0.6392	116.0510	6.6572	6.6573	6.9231	11.4255	20.3365
16	Vtub	35.2724	6.0531	0.9476	2.4728	0.6598	101.2283	4.6866	4.7335	5.2965	8.1311	18.4415
17	TC	50.7521	7.5092	0.9245	21.0460	1.1721	121.4836	5.5793	5.6745	6.5706	9.7535	19.7150
18	3P	40.5560	6.8976	0.9397	4.5115	0.8180	107.3639	5.0151	5.0748	6.0354	8.7189	18.9300
19	DP1	319.4078	17.5031	0.5477	219.278	2.8473	190.4276	14.6847	16.8565	15.9119	29.2453	30.1207
20	DP2	4668.094	73.3747	−5.7994	8.23×10^{11}	11.9998	5,039.181	28.0112	56.3689	65.2219	99.9997	37.6722
21	DPF	19.0466	4.3652	0.9722	0.1630	0.1495	92.2767	3.7212	3.7218	3.8802	6.3876	15.6657
22	Proposed model	18.9722	4.3544	0.9723	0.1615	0.1482	92.2155	3.7139	3.7145	3.8706	6.3751	15.6500

3.5. Results of Dataset 3

Table 7 shows the estimated values for the parameters of each model obtained using dataset 3. Each parameter of the proposed model is represented through $\hat{a} = 194.7684$, $\hat{b} = 0.3062$, $\hat{c} = 307.0805$, and $\hat{h} = 135.5641$. Figure 4 shows the results of calculating the estimated value of $m(t)$ at each point in time based on the number of cumulative failures at each time point in dataset 3 and for each model equation. The black dotted line indicates the actual data, and the dark red solid line is the predicted failure value at each time point for the proposed model. Compared with other models, the proposed model shows the predicted value closest to the actual value.

Table 8 shows the results of calculating the criteria of each model using the parameters obtained through dataset 3. As a result, MSE and PC of the proposed model show the smallest values of 26.8047 and 24.5551, respectively, and Adj_R^2 shows the closest value to 1 at 0.9765. In addition, MAE, PRR, PP, PRV, RMSPE, MEOP, and TS are 4.9209, 0.0096, 0.0092, 4.6668, 4.6668, 4.5694, and 2.5484, respectively, showing the second smallest values. Figure 4 shows the estimated failure values at each time point using the developed models. The Vtub model shows the most suitable criteria of 0.0094, 0.0090, 4.6356, 4.6357, and 2.5315 in PRR, PP, PRV, RMSPE, and TS, and DPF shows the most suitable criteria of 4.9195 and 4.5682 with MAE and MEOP. However, in calculating the AIC of the Vtub model, DPF, KSRGM, and the newly proposed model, a value indicating $t = 14$ is shown, indicating that the calculation is no longer being applied. In the process of calculating the AIC value, if there is no difference between the value at a specific point in time and the next point in time, the denominator is 0, so the AIC calculation can not be performed.

Table 7. Parameter estimation of model from dataset 3.

No.	Model	Estimation
1	GO	$\hat{a} = 197.387, \hat{b} = 0.399$
2	HDOG	$\hat{a} = 197.3858, \hat{b} = 0.3985, \hat{c} = 0.00088$
3	DS	$\hat{a} = 192.528, \hat{b} = 0.882$
4	IS	$\hat{a} = 197.354, \hat{b} = 0.399, \hat{\beta} = 0.000001$
5	ZFR	$\hat{a} = 198.0864, \hat{b} = 0.0038, \hat{\alpha} = 1545.538$ $\hat{\beta} = 3.7206, \hat{c} = 603.6647, \hat{p} = 4.7236$
6	YE	$\hat{a} = 248.808, \hat{\alpha} = 0.00797$ $\hat{\beta} = 0.2253, \hat{\gamma} = 208.4032$
7	YR	$\hat{a} = 206.0833, \hat{\alpha} = 1.0937$ $\hat{\beta} = 0.1427, \hat{\gamma} = 2.3984$
8	YID1	$\hat{a} = 183.4522, \hat{b} = 0.4620, \hat{\alpha} = 0.0066$
9	YID2	$\hat{a} = 182.934, \hat{b} = 0.464, \hat{\alpha} = 0.0071$
10	PZ	$\hat{a} = 195.990, \hat{b} = 0.3987, \hat{\alpha} = 1000.0, \hat{\beta} = 0.0000,$ $\hat{c} = 1.390$
11	PNZ	$\hat{a} = 183.125, \hat{b} = 0.463$ $\hat{\alpha} = 0.007, \hat{\beta} = 0.0001$
12	TP	$\hat{a} = 21.2071, \hat{b} = 0.3086, \hat{\alpha} = 0.9415, \hat{\beta} = 0.0209$ $\hat{c} = 1.8073, \hat{p} = 0.2950, \hat{q} = 0.1959$
13	KSRGM	$\hat{A} = 61.8904, \hat{b} = 53.2238$ $\hat{\alpha} = 0.6853, \hat{p} = 0.0256$
14	RMD	$\hat{a} = 5.4873, \hat{b} = 0.3986$ $\hat{\alpha} = 35.9711, \hat{\beta} = 208.4596$
15	IFD	$\hat{a} = 23.5220, \hat{b} = 0.6188, \hat{d} = 0.000000002$
16	Vtub	$\hat{a} = 1.2170, \hat{b} = 2.9515, \hat{\alpha} = 0.0595$ $\hat{\beta} = 0.000006, \hat{N} = 194.7808$
17	TC	$\hat{a} = 0.053, \hat{b} = 0.774, \hat{\alpha} = 181.0$ $\hat{\beta} = 38.600, \hat{N} = 204.140$
18	3P	$\hat{a} = 0.0307, \hat{b} = 0.2038, \hat{\beta} = 0.000581$ $\hat{N} = 203.8418, \hat{c} = 100.8152$
19	DP1	$\hat{\alpha} = 0.1105, \hat{\beta} = 3.1046$
20	DP2	$\hat{\alpha} = 0.0290, \hat{\gamma} = 6.0596$ $\widehat{t_0} = 0.7697, \widehat{m_0} = 0.0387$
21	DPF	$\hat{a} = 194.766, \hat{b} = 0.304$ $\hat{c} = 304.566, \hat{h} = 135.464$
22	Proposed model	$\hat{a} = 194.7684, \hat{b} = 0.3062$ $\hat{c} = 307.0805, \hat{h} = 135.5641$

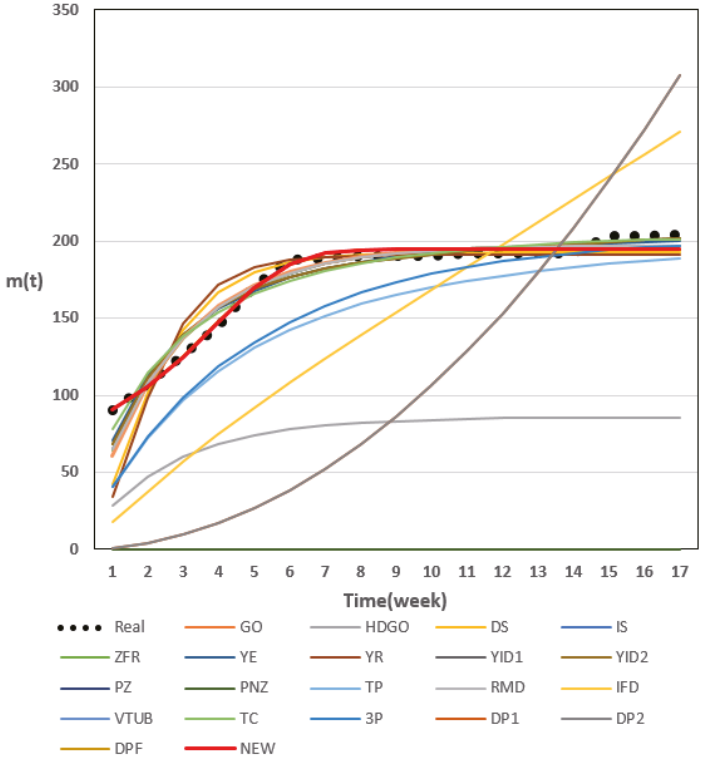


Figure 4. Prediction of all models for dataset 3.

Table 8. Comparison of all criteria from dataset 3.

No.	Model	MSE	MAE	Adj_R^2	PRR	PP	AIC	PRV	RMSPE	MEOP	TS	PC
1	GO	80.6779	6.9602	0.9301	0.1705	0.1013	184.3314	8.6734	8.6955	6.5252	4.7492	34.1231
2	HDOG	86.4370	7.4526	0.9247	0.1714	0.1015	186.2332	8.6705	8.6951	6.9558	4.7491	33.2854
3	DS	232.628	9.5029	0.7982	1.2915	0.3330	331.8567	14.6423	14.7605	8.9090	8.0644	42.0654
4	IS	86.4395	7.4550	0.9247	0.1706	0.1013	186.3337	8.6711	8.6953	6.9580	4.7492	33.2856
5	ZFR	111.138	9.4352	0.9010	0.1837	0.1047	193.0813	8.7023	8.7388	8.6489	4.7734	32.2423
6	YE	79.3698	8.1539	0.9304	0.0993	0.0738	167.3203	8.0202	8.0298	7.5715	4.3853	31.6111
7	YR	378.666	12.8708	0.6679	2.7655	0.4685	523.3189	17.3551	17.5296	11.9514	9.5785	41.7676
8	YID1	78.8312	7.1635	0.9313	0.1282	0.0868	157.6388	8.2937	8.3046	6.6860	4.5353	32.6406
9	YID2	78.8367	7.1869	0.9313	0.1276	0.0866	157.8252	8.2915	8.3047	6.7078	4.5355	32.6411
10	PZ	100.990	8.6962	0.9108	0.1719	0.1017	190.3321	8.6767	8.7014	8.0272	4.7525	32.2669
11	PNZ	84.9077	7.7388	0.9256	0.1281	0.0867	159.8744	8.2915	8.3049	7.1860	4.5356	32.0493
12	TP	98.6971	10.573	0.9113	0.0740	0.0626	166.0911	7.8528	7.8540	9.6118	4.2889	31.5071
13	KSRGM	111.132	8.1738	0.9025	0.2521	0.1297	NA	9.4631	9.5000	7.5900	5.1890	33.7990
14	RMD	93.1051	8.0261	0.9184	0.1714	0.1015	188.2567	8.6714	8.6960	7.4528	4.7496	32.6486
15	IFD	3691.538	60.2705	−2.2183	24.7762	2.5036	466.1166	51.3584	56.5265	56.2524	31.0359	59.5661
16	Vtub	28.6529	5.2313	0.9747	0.0094	0.0090	Inf	4.6356	4.6357	4.8289	2.5315	24.7084
17	TC	72.2812	8.5966	0.9361	0.0521	0.0479	158.9319	7.3621	7.3628	7.9353	4.0207	30.2602
18	3P	81.0554	8.8835	0.9284	0.0731	0.0614	164.5417	7.7935	7.7967	8.2001	4.2577	30.9476
19	DP1	11,068.48	101.169	−8.6006	9,218.87	6.8842	1224.361	78.7966	100.6555	94.8460	55.6271	71.0335
20	DP2	12,760.68	116.690	−10.191	11,546.76	6.8801	1248.593	78.7816	100.6144	108.3548	55.6039	64.6312
21	DPF	26.8104	4.9195	0.9765	0.0096	0.0092	NA	4.6673	4.6673	4.5682	2.5487	24.5565
22	Proposed model	26.8047	4.9209	0.9765	0.0096	0.0092	NA	4.6668	4.6668	4.5694	2.5484	24.5551

4. Optimal Release Time

When releasing software, it is very important that find the optimal release time. In order to find that, we need to find a time that minimizes the cost. We apply $m(t)$ proposed in Section 2 to the cost model to find the optimal time point between time to market and the minimum cost. The optimal time is suggested based on the cost model that reflects the software installation cost, software test cost, operation cost, software removal cost, and risk cost when the software failure occurs. Figure 5 describes the software field environment from the software installation of the software cost model. The expected software cost model follows Equation (22) [30,31].

$$C(T) = C_0 + C_1 T + C_2 m(T) + C_3(1 - R(x|T)) \tag{22}$$

where C_0 is the installation cost for system testing, C_1 is the system test cost per unit time, C_2 is the error removal cost per unit time during the test phase, and C_3 is the penalty cost owing to a system failure. In addition, x represents the time the software was used. In addition, in the cost model equation, $R(x|T)$ follows (23) [32,33].

$$R(x|T) = e^{-[m(t+x)-m(t)]} \tag{23}$$

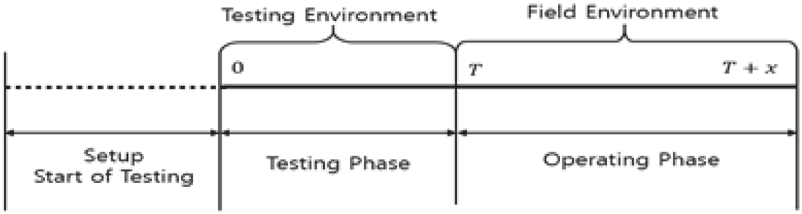


Figure 5. System cost model structure.

In this section, we propose a cost model using dataset 1 based on the proposed software reliability model and find the optimal time point between time to market and the minimum cost by changing the cost coefficients from C_0 to C_3.

4.1. Results of the Optimal Release Time

For the parameters of the cost model, a, b, c, and h calculated through numerical examples described in Section 3 were used. The cost coefficient of the cost model aims to find the optimal release time with the lowest cost by finding the optimal value through the changes in several values. The baseline value of the cost coefficient is as follows:

$$C_0 = 500,\ C_1 = 20,\ C_2 = 50,\ C_3 = 5000,\ x = 6$$

Here, baseline denotes to the reference value for confirming the change of the cost coefficient. The total cost value obtains as a reference value is 4888.856, and the optimal release time T at this time is 18.3. Table 9 changes the cost coefficient of each reference value, checks the minimum cost $C(T)$ and optimal release time T^*, and then checks the changing trend to find the most optimal release time T^*. When $x = 2$, the smallest total cost value obtains 4886.985 at $T^* = 18.2$. When $x = 4$, the smallest total cost value shows 4888.735 at $T^* = 18.3$. When $x = 6$, the smallest total cost value shows 4888.856 at $T^* = 18.3$. When x is 8 and 10, the smallest total cost value shows 4888.863 at $T^* = 18.3$.

Table 9. Optimal release time of expected total cost according to baseline.

Base	***x* = 2**		***x* = 4**		***x* = 6**		***x* = 8**		***x* = 10**	
	T*	***C*(*T*)**	***T****	***C*(*T*)**	***T****	***C*(*T*)**	***T****	***C*(*T*)**	***T****	***C*(*T*)**
	18.2	4886.985	18.3	4888.735	18.3	4888.856	18.3	4888.863	18.3	4888.863

Here, C_0 is the setup cost, and as the value increases, the cost, which is directly proportional, increases as well; thus, the lower the setup cost is, the lower the cost. Table 10 compares the changes when the coefficients of are 300, 500, and 700. It is found that the higher the value is, the higher the total cost value, whereas the optimal time does not change. Therefore, it appears that C_0 does not help determine the optimal release point. However, because the setup cost for a system stabilization is required, the appropriate C_0 cost coefficient is set to 500. Figure 6 shows a graph of the results according to the change in C_0.

Table 10. Optimal release time of expected total cost according to C_0.

C_0	***x* = 2**		***x* = 4**		***x* = 6**		***x* = 8**		***x* = 10**	
	T*	***C*(*T*)**	***T****	***C*(*T*)**	***T****	***C*(*T*)**	***T****	***C*(*T*)**	***T****	***C*(*T*)**
300	18.2	4686.985	18.3	4688.735	18.3	4688.856	18.3	4688.863	18.3	4688.863
500	18.2	4886.985	18.3	4888.735	18.3	4888.856	18.3	4888.863	18.3	4888.863
700	18.2	5086.985	18.3	5088.735	18.3	5088.856	18.3	5088.863	18.3	5088.863

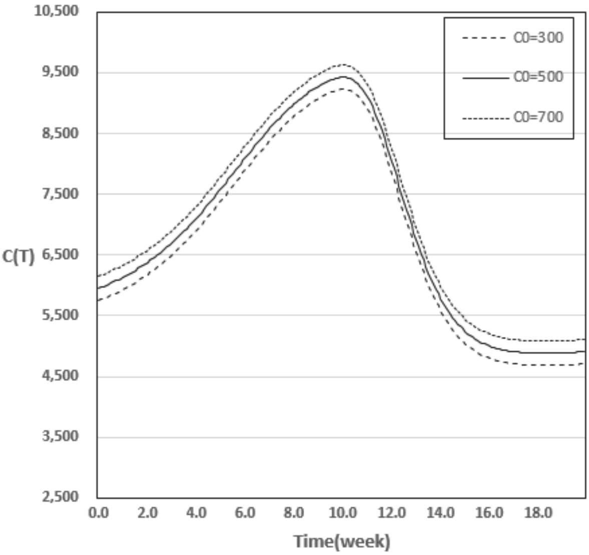


Figure 6. Optimal release time of total cost according to C_0.

Table 11 compares the changes when the coefficients of C_1 are 10, 20, and 30. The results show that when C_1 is 10, the total cost is the minimum at approximately 18.9 to 19.0, and when C_1 is 20, the minimum value is at 18.2 to 18.3, and when it is 30, the total cost shows the minimum value at approximately 17.8 to 17.9. As the cost coefficient C_1 increases, the optimal release time is gradually pushed back. Figure 7 shows a graph of the results according to the changes in C_1.

Table 11. Optimal release time of expected total cost according to C_1.

C_1	x=2		x=4		x=6		x=8		x=10	
	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$
10	18.9	4702.044	19.0	4702.818	19.0	4702.864	19.0	4702.866	19.0	4702.866
20	18.2	4886.985	18.3	4888.735	18.3	4888.856	18.3	4888.863	18.3	4888.863
30	17.8	5066.820	17.9	5069.652	17.9	5069.861	17.9	5069.873	17.9	5069.874

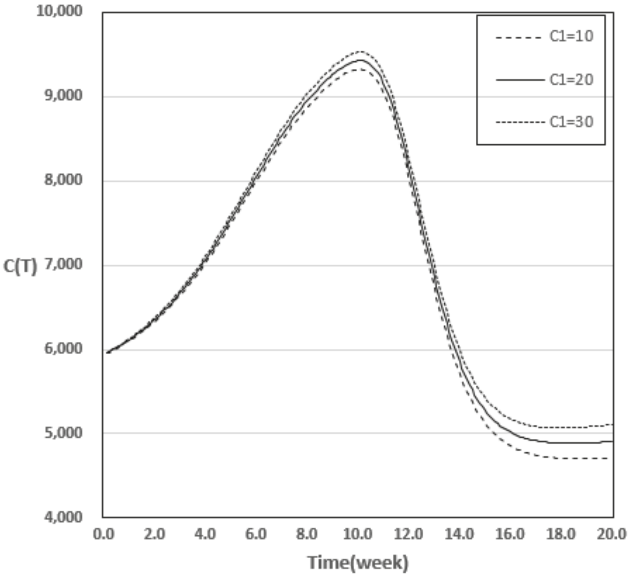


Figure 7. Optimal release time of total cost according to C_1.

Table 12 compares the changes when the coefficients of C_2 are 30, 40, 50, and 60. It can be seen that the cost coefficient C_2 does not change from 18.2 to 18.3 at the optimal release time as the value changes. Figure 8 shows a graph of the results according to the change in C_2.

Table 13 compares the changes when the coefficients of C_3 are 5000, 7000, 10,000, and 15,000. The results show that when C_3 is 5000, the total cost is the minimum at approximately 18.2 to 18.3; when it is 7000, it shows the minimum value at 18.5 to 18.6; when it is 10,000, the total cost shows the minimum value at approximately 18.9 to 19.0; and when it is 15,000, the total cost shows the minimum value at approximately 19.2 to 19.3. This indicates that the optimal release time gradually increases as the cost coefficient C_3 increases. Figure 9 shows a graph of the results according to the changes in C_3.

Table 12. Optimal release time of expected total cost according to C_2.

C_2	x=2		x=4		x=6		x=8		x=10	
	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$
30	18.2	3285.254	18.3	3286.995	18.3	3287.116	18.3	3287.123	18.3	3287.123
40	18.2	4086.120	18.3	4087.865	18.3	4087.986	18.3	4087.993	18.3	4087.993
50	18.2	4886.985	18.3	4888.735	18.3	4888.856	18.3	4888.863	18.3	4888.863
60	18.2	5687.851	18.3	5689.604	18.3	5689.726	18.3	5689.733	18.3	5689.733

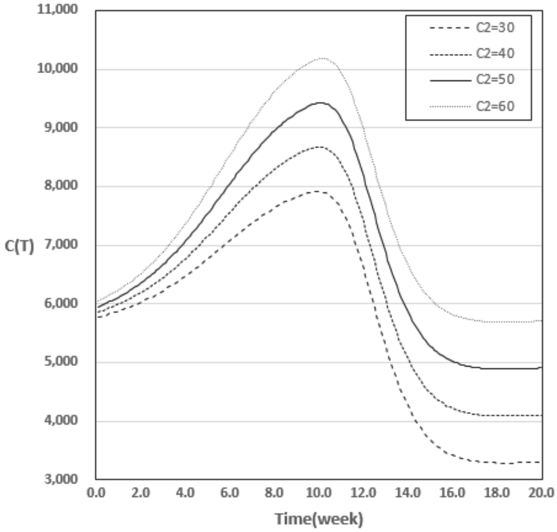


Figure 8. Optimal release time of the total cost according to C_2.

Table 13. Optimal release time of expected total cost according to C_3.

C_3	x=2		x=4		x=6		x=8		x=10	
	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$
5000	18.2	4886.985	18.3	4888.735	18.3	4888.856	18.3	4888.863	18.3	4888.863
7000	18.5	4893.210	18.6	4894.846	18.6	4894.958	18.6	4894.964	18.6	4894.964
10,000	18.9	4899.648	19.0	4901.186	19.0	4901.277	19.0	4901.281	19.0	4901.282
15,000	19.2	4906.802	19.3	4908.208	19.3	4908.297	19.3	4908.301	19.3	4908.301

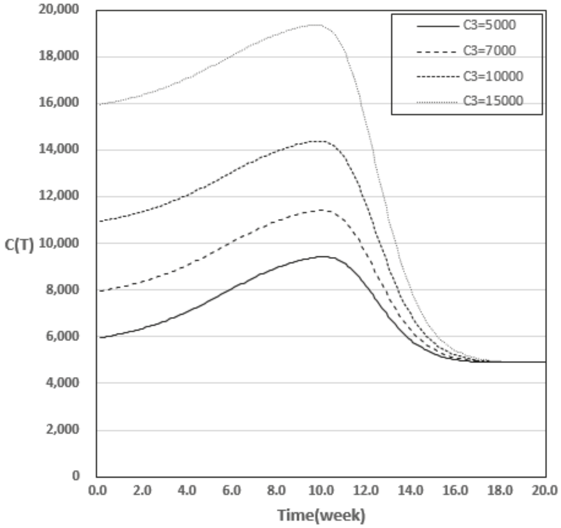


Figure 9. Optimal release time of the total cost according to C_3.

4.2. Results of Variation in Cost Model for Changes in Parameter

In this section, we check whether the optimal release time is affected by the change in the cost model according to the change in the parameters of the proposed model. The parameters a, b, c, and h of the proposed model are set at −20%, −10%, 0%, 10%, and 20%, respectively, in 10% increments, and the coefficient of the cost model is fixed at the baseline value in Section 4.1. Thus, the minimum cost value is calculated depending on changes in the parameters, and it derives appropriate release time. In Table 14, 0% is the same as the value suggested in Table 9 by substituting the parameter estimates described in Section 3 and the coefficient values of the cost model proposed in Section 4.

Table 14. Optimal release time of cost according to parameter change.

	−20%		−10%		0%		10%		20%	
	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$	T^*	$C(T)$
a	20.4	4129.849	19.2	4507.887	18.3	4888.856	17.6	5272.215	16.8	5657.258
b	22.6	4979.801	20.0	4929.426	18.3	4888.856	16.8	4855.545	15.4	4827.546
c	16.4	4847.806	17.4	4869.089	18.3	4888.856	19.2	4907.319	20.0	4924.648
h	18.6	4892.931	18.4	4890.757	18.3	4888.856	18.2	4887.130	18.2	4885.583

From Table 14 and Figures 10–13, the value of the cost model $C(T)$ increases as the change in parameter a increases, whereas the optimal release time T^* decreases. As the values of parameters b and h increase, the cost model $C(T)$ increases, and the release time T^* is shown to decrease; in addition, it is found that the change in parameter h had a very slight effect on the optimal release time compared to parameter b. As the value of parameter c increases, the cost model $C(T)$ and release time T* increase together. Based on this, it is found that parameter a had a very large minimum width of the cost model compared with the changes of the other parameters, and parameter b had the greatest influence on determining the optimal release time.

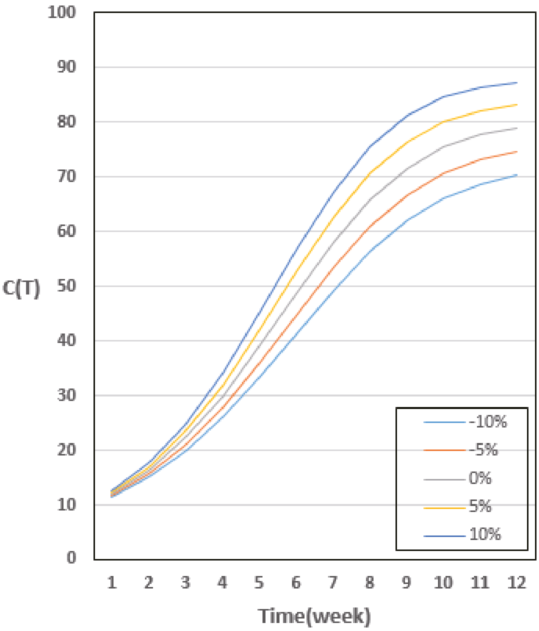


Figure 10. Optimal release time of cost according to a.

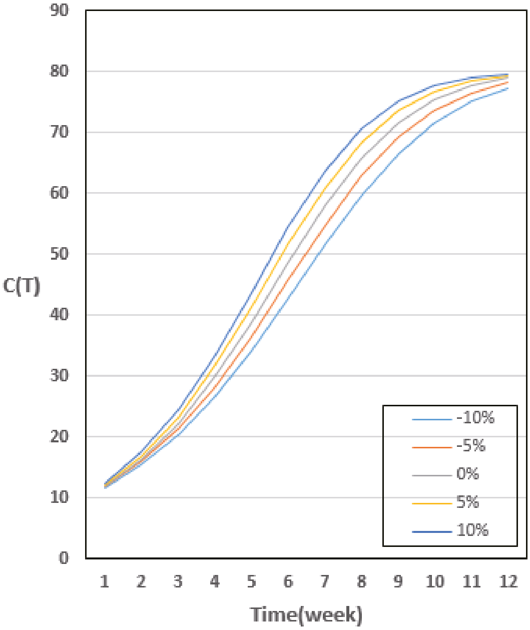


Figure 11. Optimal release time of cost according to *b*.

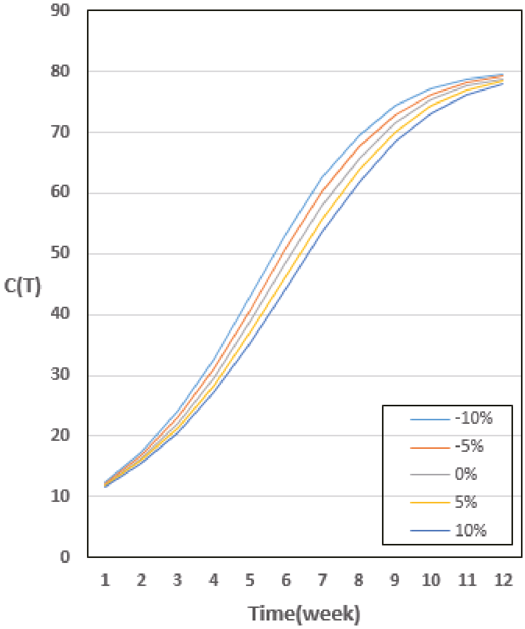


Figure 12. Optimal release time of cost according to *c*.

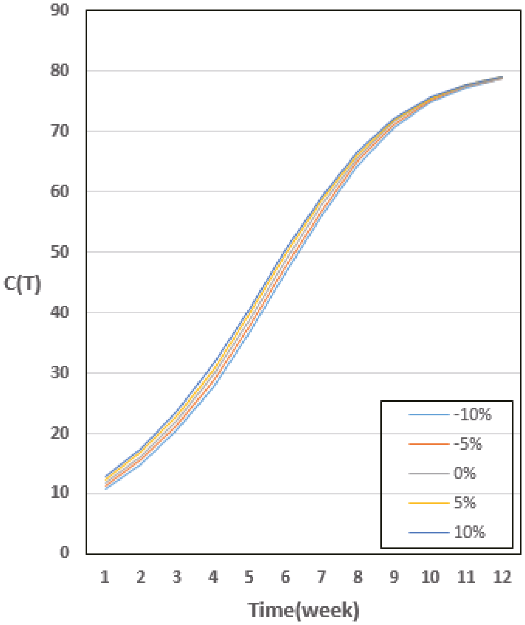


Figure 13. Optimal release time of cost according to *h*.

5. Conclusions

In this study, a new software reliability model was developed under the assumption that software failures occur in a dependent manner. We used three datasets for our evaluations. The first and second datasets showed the best fit, and the third dataset showed better results compared with many previously proposed models. The proposed model showed better results than DP1, DP2, and DPF, which are previously developed software-dependent failure occurrence models.

In addition, based on the proposed model, the optimal release time according to the change in the cost coefficient was suggested, and the total cost was analyzed accordingly. When the test cost was increased, the release time gradually increased, as did the overall cost; therefore, the optimal release time can be achieved when C_1 is 20. In the proposed model, fault detection rate *b* was found to be the most important parameter for determining the optimal release time.

In the past, studies were conducted by assuming independence in the case of software failures; however, in a real environment, the software execution environment is extremely diverse and complex. Therefore, it is necessary to develop a model that assumes a dependent failure occurrence and propose a model that considers the actual operating environment. We plan to conduct a study using machine learning and deep learning for the proposed software-dependent failure occurrence in the future work.

Author Contributions: Conceptualization, H.P.; Funding acquisition, I.H.C., K.Y.S.; Software, Y.S.K.; Writing—original draft, Y.S.K.; Writing—review and editing, K.Y.S., I.H.C. and H.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1B07045734, NRF-2021R1F1A1048592, and NRF-2021R1I1A1A01059842).

Institutional Review Board Statement: Not required.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository.

Acknowledgments: This research was supported by the National Research Foundation of Korea. We are pleased to thank the Editor and the Referees for their useful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wu, Y.C.; Wu, Y.J.; Wu, S.M. An outlook of a future smart city in Taiwan from post–internet of things to artificial intelligence internet of things. In *Smart Cities: Issues and Challenges*; Elsevier: Amsterdam, The Netherlands, 2019; pp. 263–282.
2. Goel, A.L.; Okumoto, K. Time-dependent error-detection rate model for software reliability and other performance measures. *IEEE Trans. Reliab.* **1979**, *28*, 206–211. [CrossRef]
3. Hossain, S.A.; Dahiya, R.C. Estimating the parameters of a non-homogeneous Poisson-process model for software reliability. *IEEE Trans. Reliab.* **1993**, *42*, 604–612. [CrossRef]
4. Yamada, S.; Ohba, M.; Osaki, S. S-shaped reliability growth modeling for software fault detection. *IEEE Trans. Reliab.* **1983**, *32*, 475–484. [CrossRef]
5. Ohba, M. Inflection S-shaped software reliability growth model. In *Stochastic Models in Reliability Theory*; Osaki, S., Hatoyama, Y., Eds.; Springer: Berlin, Germany, 1984; pp. 144–162.
6. Zhang, X.M.; Teng, X.L.; Pham, H. Considering fault removal efficiency in software reliability assessment. *IEEE Trans. Syst. Man Cybern. Part A-Syst. Hum.* **2003**, *33*, 114–120. [CrossRef]
7. Yamada, S.; Ohtera, H.; Narihisa, H. Software Reliability Growth Models with Testing-Effort. *IEEE Trans. Reliab.* **1986**, *35*, 19–23. [CrossRef]
8. Yamada, S.; Tokuno, K.; Osaki, S. Imperfect debugging models with fault introduction rate for software reliability assessment. *Int. J. Syst. Sci.* **1992**, *23*, 2241–2252. [CrossRef]
9. Pham, H.; Zhang, X. An NHPP software reliability models and its comparison. *Int. J. Reliab. Qual. Saf. Eng.* **1997**, *4*, 269–282. [CrossRef]
10. Pham, H.; Nordmann, L.; Zhang, X. A general imperfect software debugging model with S-shaped fault detection rate. *IEEE Trans. Reliab.* **1999**, *48*, 169–175. [CrossRef]
11. Teng, X.; Pham, H. A new methodology for predicting software reliability in the random field environments. *IEEE Trans. Reliab.* **2006**, *55*, 458–468. [CrossRef]
12. Kapur, P.K.; Pham, H.; Anand, S.; Yadav, K. A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation. *IEEE Trans. Reliab.* **2011**, *60*, 331–340. [CrossRef]
13. Roy, P.; Mahapatra, G.S.; Dey, K.N. An NHPP software reliability growth model with imperfect debugging and error generation. *Int. J. Reliab. Qual. Saf. Eng.* **2014**, *21*, 1–3. [CrossRef]
14. Pham, H. *System Software Reliability*; Springer: London, UK, 2006.
15. Pham, H. A new software reliability model with Vtub-Shaped fault detection rate and the uncertainty of operating environments. *Optimization* **2014**, *63*, 1481–1490. [CrossRef]
16. Chang, I.H.; Pham, H.; Lee, S.W.; Song, K.Y. A testing-coverage software reliability model with the uncertainty of operation environments. *Int. J. Syst. Sci.-Oper. Logist.* **2014**, *1*, 220–227.
17. Song, K.Y.; Chang, I.H.; Pham, H. A Three-parameter fault-detection software reliability model with the uncertainty of operating environments. *J. Syst. Sci. Syst. Eng.* **2017**, *26*, 121–132. [CrossRef]
18. Ramasamy, S.; Lakshmanan, I. Machine learning approach for software reliability growth modeling with infinite testing effort function. *Math. Probl. Eng.* **2017**, 8040346. [CrossRef]
19. Kim, Y.S.; Chang, I.H.; Lee, D.H. Non-Parametric Software Reliability Model Using Deep Neural Network and NHPP Software Reliability Growth Model Comparison. *J. Korean. Data Anal. Soc.* **2020**, *22*, 2371–2382. [CrossRef]
20. Begum, M.; Hafiz, S.B.; Islam, J.; Hossain, M.J. Long-term Software Fault Prediction with Robust Prediction Interval Analysis via Refined Artificial Neural Network (RANN) Approach. *Eng. Lett.* **2021**, *29*, 1158–1171.
21. Zhu, J.; Gong, Z.; Sun, Y.; Dou, Z. Chaotic neural network model for SMISs reliability prediction based on interdependent network SMISs reliability prediction by chaotic neural network. *Qual. Reliab. Eng. Int.* **2021**, *37*, 717–742. [CrossRef]
22. Sahu, K.; Alzahrani, F.A.; Srivastava, R.K.; Kumar, R. Evaluating the Impact of Prediction Techniques: Software Reliability Perspective. *CMC-Comput. Mat. Contin.* **2021**, *67*, 1471–1488. [CrossRef]
23. Ogundoyin, S.O.; Kamil, I.A. A Fuzzy-AHP based prioritization of trust criteria in fog computing services. *Appl. Soft. Comput.* **2020**, *97*, 106789. [CrossRef]
24. Rafi, S.; Akbar, M.A.; Yu, W.; Alsanad, A.; Gumaei, A.; Sarwar, M.U. Exploration of DevOps testing process capabilities: An ISM and fuzzy TOPSIS analysis. *Appl. Soft Comput.* **2021**, 108377. [CrossRef]
25. Li, Q.; Pham, H. Modeling Software Fault-Detection and Fault-Correction Processes by Considering the Dependencies between Fault Amounts. *Appl. Sci.* **2021**, *11*, 6998. [CrossRef]
26. Son, H.I.; Kwon, K.R.; Kim, J.O. Reliability Analysis of Power System with Dependent Failure. *J. Korean. Inst. Illum. Electr. Install. Eng.* **2011**, *25*, 62–68.

27. Pan, Z.; Nonaka, Y. Importance analysis for the systems with common cause failures. *Reliab. Eng. Syst. Saf.* **1995**, *50*, 297–300. [CrossRef]
28. Pham, L.; Pham, H. Software Reliability Models with Time Dependent Hazard Rate Based on Bayesian Approach. *IEEE Trans. Syst. Man Cybern. Part A-Syst. Hum.* **2000**, *30*, 25–35. [CrossRef]
29. Lee, D.H.; Chang, I.H.; Pham, H. Software reliability model with dependent failures and SPRT. *Mathematics* **2020**, *8*, 1366. [CrossRef]
30. Kim, H.C. The Property of Learning effect based on Delayed Software S-Shaped Reliability Model using Finite NHPP Software Cost Model. *Indian J. Sci. Technol.* **2015**, *8*, 1–7. [CrossRef]
31. Yang, B.; Xie, M. A study of operational and testing reliability in software reliability analysis. *Reliab. Eng. Syst. Saf.* **2000**, *70*, 323–329. [CrossRef]
32. Yamada, S.; Osaki, S. Cost-reliability optimal release policies for software systems. *IEEE Trans. Reliab.* **1985**, *34*, 422–424. [CrossRef]
33. Singpurwalla, N.D. Determining an optimal time interval for testing and debugging software. *IEEE Trans. Softw. Eng.* **1991**, *17*, 313–319. [CrossRef]
34. Song, K.Y.; Chang, I.H. A Sensitivity Analysis of a New NHPP Software Reliability Model with the Generalized Exponential Fault Detection Rate Function Considering the Uncertainty of Operating Environments. *J. Korean. Data Anal. Soc.* **2020**, *22*, 473–482.
35. Li, X.; Xie, M.; Ng, S.H. Sensitivity analysis of release time of software reliability models incorporating testing effort with multiple change-points. *Appl. Math. Model.* **2010**, *34*, 3560–3570. [CrossRef]
36. Stringfellow, C.; Andrews, A.A. An empirical method for selecting software reliability growth models. *Empir. Softw. Eng.* **2002**, *7*, 319–343. [CrossRef]
37. Inoue, S.; Yamada, S. Discrete software reliability assessment with discretized NHPP models. *Comput. Math. Appl.* **2006**, *51*, 161–170. [CrossRef]
38. Anjum, M.; Haque, M.A.; Ahmad, N. Analysis and ranking of software reliability models based on weighted criteria value. *Int. J. Inf. Technol. Comput. Sci.* **2013**, *2*, 1–14. [CrossRef]
39. Jeske, D.R.; Zhang, X. Some successful approaches to software reliability modeling in industry. *J. Syst. Softw.* **2005**, *74*, 85–99. [CrossRef]
40. Iqbal, J. Software reliability growth models: A comparison of linear and exponential fault content functions for study of imperfect debugging situations. *Cogent Eng.* **2017**, *4*, 1286739. [CrossRef]
41. Zhao, J.; Liu, H.W.; Cui, G.; Yang, X.Z. Software reliability growth model with change-point and environmental function. *J. Syst. Softw.* **2006**, *79*, 1578–1587. [CrossRef]
42. Akaike, H. A new look at statistical model identification. *IEEE Trans. Autom. Control* **1974**, *19*, 716–719. [CrossRef]
43. Pillai, K.; Nair, V.S. A model for software development effort and cost estimation. *IEEE Trans. Softw. Eng.* **1997**, *23*, 485–497. [CrossRef]
44. Peng, R.; Li, Y.F.; Zhang, W.J.; Hu, Q.P. Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction. *Reliab. Eng. Syst. Saf.* **2014**, *126*, 37–43. [CrossRef]
45. Sharma, K.; Garg, R.; Nagpal, C.K.; Garg, R.K. Selection of optimal software reliability growth models using a distance based approach. *IEEE Trans. Reliab.* **2010**, *59*, 266–276. [CrossRef]
46. Pham, H. On estimating the number of deaths related to Covid-19. *Mathematics* **2020**, *8*, 655. [CrossRef]
47. Wang, L.; Hu, Q.; Liu, J. Software reliability growth modeling and analysis with dual fault detection and correction processes. *IIE Trans.* **2016**, *48*, 359–370. [CrossRef]

symmetry

MDPI

Article

Sigma Identification Protocol Construction Based on MPF

Eligijus Sakalauskas [1,*], Inga Timofejeva [2] and Ausrys Kilciauskas [3]

[1] Department of Applied Mathematics, Kaunas University of Technology, K. Donelaicio Str. 73, 44249 Kaunas, Lithuania
[2] Center for Nonlinear Systems, Kaunas University of Technology, Studentu 50-147, 51368 Kaunas, Lithuania; inga.timofejeva@ktu.lt
[3] Department of Applied Informatics, Vytautas Magnus University, K. Donelaičio Str. 58, 44248 Kaunas, Lithuania; ausrys.kilciauskas@vdu.lt
* Correspondence: eligijus.sakalauskas@ktu.lt

Abstract: A new sigma identification protocol (SIP) based on matrix power function (MPF) defined over the modified medial platform semigroup and power near-semiring is proposed. It is proved that MPF SIP is resistant against direct and eavesdropping attacks. Our security proof relies on the assumption that MPF defined in the paper is a candidate for one-way function (OWF). Therefore, the corresponding MPF problem is reckoned to be a difficult one. This conjecture is based on the results demonstrated in our previous studies, where a certain kind of MPF problem was proven to be NP-complete.

Keywords: matrix power function; sigma identification protocol; security against eavesdropping attack; candidate for one-way function

Citation: Sakalauskas, E.; Timofejeva, I.; Kilciauskas, A. Sigma Identification Protocol Construction Based on MPF. *Symmetry* **2021**, *13*, 1683. https://doi.org/10.3390/sym13091683

Academic Editors: Ming-Chin Chuang and Jeng-Shyang Pan

Received: 21 July 2021
Accepted: 9 September 2021
Published: 13 September 2021

1. Introduction

In this paper a new paradigm for the so-called Sigma Identification Protocol (SIP) based on the authors' earlier proposed new candidate for one-way function (OWF) is presented. In general, Sigma protocols are three-round protocols similar to the well-known Schnorr identification protocol. They are typically used as sub-protocols in more complicated settings and for more advanced use. For example, Sigma protocols can easily be transformed into corresponding identification and signature schemes. Another application is to design protocols that allow one party to prove to another that certain facts are true (without revealing private information). For example, to prove that encrypted value V lies in a certain range without revealing any other information about V. Sigma protocols can be combined to make new Sigma protocols. For example, in the AND-proof construction, a Prover can convince a Verifier that he knows witnesses for a pair of statements. In the OR-proof construction, a Prover can convince a Verifier that he knows witnesses for one of two statements. These examples convince us that the development of Sigma protocols based on new paradigms is promising.

The construction of cryptographic primitives based on matrix power function (MPF) belongs to the field of so called non-commuting cryptography [1], [2]. The development of non-commuting cryptography is important due to the need to replace traditional cryptographic methods vulnerable to quantum cryptanalysis. Peter W. Shor has proposed the polynomial-time quantum cryptanalysis [3] for the traditional cryptographic primitives such as Diffie—Hellman key exchange protocol, RSA and ElGamal cryptosystems, Digital signature algorithm (DSA) and Elliptic Curve DSA (ECDSA).

One of the promising trends is the creation of OWFs, the security of which relies on the NP-hard problems [4]. Thus far, there are no known effective quantum cryptanalytic algorithms solving NP-hard problems; therefore, this cryptographic trend is a significant part of the so-called post-quantum cryptography [5]. One of the trends to create cryptographic primitives that can resist quantum cryptanalysis attacks is lattice-based cryptography [6] and hidden field equations (HFE) based cryptosystems [7–10]. Despite some cryptanalytic attacks on the HFE cryptosystem [8,9], this trend is viewed as promising [10].

MPF is somewhat related to the multivariate polynomials used in HFE cryptosystems since the complexity of so called MPF problem and the NP-completeness of this problem is proved using polynomial-time reduction of multivariate quadratic (MQ) problem [7]. Referencing [11] it is proved that certain kinds of MPF problems are NP-complete as well [12,13].

In this paper the novel MPF based sigma identification protocol (SIP) is presented using several specifically selected algebraic structures introduced in [14]. The concept of abstract MPF as well as main definitions for the construction of SIP are given in Section 2. The algebraic structures for the construction of MPF are defined in Section 3. MPF SIP is presented in Section 4. The security of MPF SIP against eavesdropping attacks is proven in Section 5. In Section 6, the selection of security parameters as well as the efficiency analysis are presented. The discussions and conclusions are presented in Section 7. The table of notations used in this paper as well as the numerical example are displayed in the Abbreviations and Appendix A, respectively.

2. The Construction of the Abstract Matrix Power Function

In this section, the matrix power function (MPF) is constructed in an abstract form without specifying exact algebraic structures that will be introduced in subsequent sections. Let $X = \{x_{il}\}$, $W = \{w_{lj}\}$ and $Y = \{y_{jk}\}$ be $m \times m$ matrices over some semiring, and indices i, j, k, $l \in I_m = \{1, 2, \ldots, m\}$. Multiplying matrix W by matrix X from the left and by matrix Y from the right yields a new matrix $Q = \{q_{ik}\}$.

$$XWY = Q; \sum_{j=1}^{m} \sum_{l=1}^{m} x_{il} w_{lj} y_{jk} = q_{ik}; i, j, k, l \in I_m. \tag{1}$$

Let $\boldsymbol{S}$ be some multiplicative semigroup and $\boldsymbol{R}$ some numerical semiring. In the case that MPF $\boldsymbol{S}$ is named a platform semigroup and $\boldsymbol{R}$ an exponent semiring. The exponent semiring of natural numbers with zero is denoted by $\boldsymbol{N}_0 = \{0, 1, 2, \ldots\}$. The corresponding semigroup of matrices defined over $\boldsymbol{S}$ is denoted by $\boldsymbol{M_S}$ and the semiring of power matrices defined over $\boldsymbol{R}$ is denoted by $\boldsymbol{M_R}$. Then, using an analogy with the matrix multiplication defined in (1), the left MPF, the right MPF and the left-right or simply MPF are introduced for X, $Y \in \boldsymbol{M_R}$, x_{il}, $y_{jk} \in \boldsymbol{R}$ and for $W \in \boldsymbol{M_S}$, $w_{lj} \in \boldsymbol{S}$ as follows.

Definition 1. *The left MPF corresponding to the matrix W powered by matrix X from the left with the MPF value equal to the matrix C =* $\{c_{ij}\}$ *has the following form:*

$$^{X}W = C \quad c_{ij} = \prod_{l=1}^{m} w_{lj}{}^{x_{il}}. \tag{2}$$

Definition 2. *The right MPF corresponding to the matrix W powered by matrix Y from the right with the MPF value equal to the matrix D =* $\{d_{lk}\}$ *has the following form:*

$$W^{Y} = D \quad d_{lk} = \prod_{j=1}^{m} w_{lj}{}^{y_{jk}}. \tag{3}$$

Definition 3. *The left-right, or simply MPF corresponds to the matrix W powered by matrix X from the left and by matrix Y from the right with the MPF value equal to the matrix Q =* $\{q_{ik}\}$ *and is expressed in the following way:*

$$^{X}W^{Y} = A, \quad a_{ik} = \prod_{j=1}^{m} \prod_{l=1}^{m} w_{lj}^{x_{il} \cdot y_{jk}}; i, j, k, l \in I^{(m)}. \tag{4}$$

The MPF definition is related to the following associativity identities.

Definition 4. *MPF is one-side, (left-side or right-side) associative and two-side associative if the following respective identities hold:*

$$^{Y}(^{X}W) = {}^{(YX)}W = {}^{YX}W;\ (W^{X})^{Y} = W^{(XY)} = W^{XY}. \tag{5}$$

$$(^{X}W)^{Y} = {}^{X}(W^{Y}) = {}^{X}W^{Y}. \tag{6}$$

In general, MPF is a function F: $\boldsymbol{M_R} \times \boldsymbol{M_S} \times \boldsymbol{M_R} \to \boldsymbol{M_S}$. To be concise, we will use the notation MPF_S^R for the definition of MPF with base matrix defined over the platform semigroup $\boldsymbol{S}$ in $\boldsymbol{M_S}$ and with power matrices defined over the exponent semiring $\boldsymbol{R}$ in $\boldsymbol{M_R}$. The categorical interpretation of MPF is presented in [15], in the context of the construction of several key agreement protocols. We slightly reformulate the notions used in the authors' interpretation by the following proposition, which is more appropriate for our study.

Proposition 1. *If MPF is associative, then* $\boldsymbol{M_S}$ *is a multiplicative* $\boldsymbol{M_R}$*-semibimodule.*

This means that there exist bilinear (left and right) actions of the matrix semiring $\boldsymbol{M_R}$ on the matrix semigroup $\boldsymbol{M_S}$. According to the definition of action, it must satisfy the associative law corresponding to Definition 4. Since matrix semigroup $\boldsymbol{M_S}$ is multiplicative, then $\boldsymbol{M_R}$-semibimodule $\boldsymbol{M_S}$ is multiplicative in our case. The following lemma is presented without proof. The proof can be found in [14].

Lemma 1. *If* $\boldsymbol{R}$ *is a commutative numerical semiring (e.g.,* $\boldsymbol{N}_0$ *= {0, 1, 2, ... }) and* $\boldsymbol{S}$ *is a commutative semigroup, then MPF is two-side associative.*

The direct MPF value computation requires finding matrix A in (4), when matrices X, Y and W are given. The inverse MPF value computation requires finding matrices X and Y in (4), when matrices W and A are given. The MPF problem is the computation of the inverse MPF value.

Definition 5. *A function F:* **Dom** $\to$ **Ran** *with finite sets of domain (***Dom***) and range (***Ran***) is a candidate for* ***one-way function*** *(OWF) if for all d* $\in$ **Dom** *the F(d) can be computed by a polynomial time algorithm, but any polynomial time randomized algorithm that attempts to compute an inverse value* $F^{-1}(r) = d$ *for F, where r* $\in$ **Ran** *is given, succeeds with negligible probability. That is, for all randomized algorithms, all positive integers c and all sufficiently large n = length(d), the probability to compute an inverse value r for F is at most* n^{-c}*. The probability is taken over the choice of r from the discrete uniform distribution in* **Ran***.*

Paraphrasing this definition in a non-formal way, MPF is candidate for OWF if: (1) the MPF direct value computation is easy, and (2) the MPF problem is hard.

The computation of the direct MPF value is effective and can be done by powering elements of the platform semigroup $\boldsymbol{S}$ by elements of the exponent semiring $\boldsymbol{R}$ with relatively small values (e.g., up to 5 used in this study). It is related to the matrix multiplication by the two matrices from the left and right. In this paper we present some evidence that the solution of the MPF problem is hard.

Proposition 2. *The necessary requirements for MPF for the proposed SIP are the following: (1) it is a candidate for OWF, (2) it is associative, and (3) the following distributive identity holds:*

$$^{(U+X)}W^{(\ +Y)} = {}^{U}W^{V} * {}^{U}W^{Y} * {}^{X}W^{V} * {}^{X}W^{Y}, \tag{7}$$

*where * is a Hadamard product of matrices [16].*

3. The Definition of Algebraic Structures

In order to construct a platform semigroup for MPF, the class of modified multiplicative medial semigroups [17] is used. A medial semigroup S_M has the presentation consisting of two generators a, b and relation R_M is defined in the following way:

$$\boldsymbol{S_M} = <a, b \mid R_M>, \tag{8}$$

$$R_M\text{: } w_1abw_2 = w_1baw_2, \tag{9}$$

where w_1 and w_2 are arbitrary non-empty words in $\boldsymbol{S_M}$, written in terms of generators a and b. The reason for the introduction of the medial semigroup is the existence of the following identity, based on the relation R_M, valid for all words $w_1, w_2 \in \boldsymbol{S_M}$ and any exponent $e \in \boldsymbol{N_0}$, where $\boldsymbol{N_0} = \{0, 1, 2, \dots\}$ is the semiring of natural numbers with zero:

$$(w_1w_2)^e = {w_1}^e {w_2}^e. \tag{10}$$

In order to construct a platform semigroup $\boldsymbol{S}$ for MPF in (4), two extra relations R_1 and R_2 are added to $\boldsymbol{S_M}$:

$$R_1\text{: } a^5 = a;\ R_2\text{: } b^5 = b. \tag{11}$$

These relations can be generalized for arbitrary finite exponents instead of 5, however, only relations (11) are considered in this paper for simplicity. Thus, modified medial semigroup $\boldsymbol{S}$ has the following presentation:

$$\boldsymbol{S_M} = <a, b \mid R_M, R_1, R_2>. \tag{12}$$

Note that we define S as a multiplicative, non-commuting and cancellative semigroup.

Proposition 3. *Semigroups* $\boldsymbol{S_M}$ *and* $\boldsymbol{S}$ *are transformed into monoids by introducing an empty word as a multiplicatively neutral element, denoted by 1. Then, conveniently, the following identities hold for all* w *in* $\boldsymbol{S_M}$ *and* $\boldsymbol{S}$*:*

$$w1 = 1w = w,\ w^0 = 1;\ 0 \in \boldsymbol{N_0}. \tag{13}$$

Using relation R_M in (9) any word in $\boldsymbol{S_M}$ can be transformed to the form $w = b^s a^t b^u a^v$ moving generators a, b left and right, where $s, t, u, v \in \boldsymbol{N_0}$. Let $w = b^s a^t b^u a^v$ be such word in $\boldsymbol{S_M}$. Reformulating the Theorem 12 in [14], the normal form w_{nf} of word w in the semigroup $\boldsymbol{S_M}$ is defined by the following function nf: $\boldsymbol{S_M} \rightarrow \boldsymbol{S_{M,nf}}$ and is expressed by the relation:

$$w_{nf} = \max_{t,u}(b^s a^t b^u a^v) = b^\beta a^{i_a} b^{i_b} a^\alpha = nf(w); \alpha, \beta \in \{0, 1\}; i_a, i_b \in \mathrm{N}_0. \tag{14}$$

The normal form in the modified medial semigroup $\boldsymbol{S}$ is defined by the following theorem.

Theorem 1. *The normal form* w_η *of the word* w_{nf} *in the normal form of* $\boldsymbol{S_M}$*, is represented by the function* η*:* $\boldsymbol{S_M} \rightarrow \boldsymbol{S}$ *and obtained by applying the minimization procedure of exponents* i_a*,* i_b *in (14) using the relations* R_1*,* R_2*:*

$$w_\eta = \underset{i_a, j_b}{min}\, w_{nf}(\beta, i_a, j_b, \alpha) = \underset{i_a, j_b}{min}(b^\beta a^{i_a} b^{j_b} a^\alpha) = b^\beta a^i b^j a^\alpha = \eta(w_{nf}); \alpha, \beta \in \{0, 1\}; i_a, i_b \in \mathrm{N}_0. \tag{15}$$

Since $\boldsymbol{S}$ is a multiplicative semiring, the following exponent identities hold for any generator $g \in \{a, b\}$:

$$g^i g^j = g^{i+j};\ (g^i)^j = g^{\,ij}. \tag{16}$$

Addition and multiplication tables for exponents i, j are presented in Tables 1 and 2 below.

Table 1. Addition (+) table for exponents *i*, *j*.

+	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	2	3	4	1
2	2	3	4	1	2
3	3	4	1	2	3
4	4	1	2	3	4

Table 2. Multiplication (●) table for exponents *i*, *j*.

●	**0**	**1**	**2**	**3**	**4**
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	2	4
3	0	3	2	1	4
4	0	4	4	4	4

Referencing relations R_1, R_2 the semiring $\boldsymbol{N}_0$ can be replaced by the finite semiring $\boldsymbol{N}_4$ = {0, 1, 2, 3, 4}. This semiring has an additive semigroup with index 1 and period 4. The exponent functions defined on $\boldsymbol{S}$ are determined by non-negative exponents in semiring $\boldsymbol{N}_{4.}$

We generalize these functions by introducing the "imaginary" unit ι which has some weak analogy with complex numbers in classical numerical algebra based on the imaginary unit i ($i^2 = -1$). According to this "analogy" the set of complex exponents can be introduced and denoted by $\iota{\cdot}\boldsymbol{N}_4$, where "·" denotes a formal multiplication of ι by any number in $\boldsymbol{N}_4$. According to our assumption and using the relation R_M in (9), the following properties of exponent ι are defined:

$$\iota^2 = 1, 1 \in \boldsymbol{N_4}; a^\iota = b; b^\iota = a; t + \iota{\cdot}u \neq \iota{\cdot}u + t, \tag{17}$$

where $t, u \in \boldsymbol{N}_4$. This means that elements $t + \iota{\cdot}u$ and $\iota{\cdot}u + t$ do not commute.

The algebraic structure termed a near-semiring [18] was introduced for the construction of MPF in [14]. In general, it can be defined in the following way.

Definition 6. *A near-semiring (**NSR**) is a nonempty set with two binary operations "+" and "·", such that <**NSR**; +; 0> is an additive monoid with neutral element 0, and <**NSR**; ·; 1> is a multiplicative monoid with neutral element 1, satisfying the following (two-sided) axioms for all x, y, z in **NSR**:*

$$x{\cdot}(y + z) = x{\cdot}y + x{\cdot}z, \text{ and } \quad (x + y){\cdot}z = x{\cdot}z + y{\cdot}z, \tag{18}$$

$$0 + x = x + 0 = x; 0{\cdot}x = x{\cdot}0 = 0; 1{\cdot}x = x{\cdot}1 = x. \tag{19}$$

Referencing to this definition the special type ***NSR*** required for MPF SIP construction is defined in the following way.

Definition 7. *The exponent near-semiring **NSR** consist of non-commuting additive monoid <**NSR**; +; 0> and commuting multiplicative monoid <**NSR**; ·; 1> satisfying Definition 6 and is a union of the following sets*

$$\boldsymbol{NSR} = \boldsymbol{N_4} + \iota{\cdot}\boldsymbol{N_4} + \boldsymbol{N_4} \cup \iota{\cdot}\boldsymbol{N_4} + \boldsymbol{N_4} + \iota{\cdot}\boldsymbol{N_4}, \tag{20}$$

where the set $\boldsymbol{N_4} + \iota{\cdot}\boldsymbol{N_4} + \boldsymbol{N_4}$ defines the class of elements {$t + \iota{\cdot}u + v$} and the set $\iota{\cdot}\boldsymbol{N_4} + \boldsymbol{N_4} + \iota{\cdot}\boldsymbol{N_4}$—the class {$t{\cdot}\iota + u + v{\cdot}\iota$}, where $t, u, v \in \boldsymbol{N_4}$.

The presentation of the semigroup $\boldsymbol{S}$ by relations R_M, R_1, R_2 in (9) and (11) induces certain properties and relations in ***NSR*** that can be directly verified and are presented below without the proof.

Proposition 4. *For any $x, y \in \boldsymbol{NSR}$, where $x = t + \iota\cdot u + v$ and $y = \iota\cdot t + u + \iota\cdot v$ and $t, u, v \in \boldsymbol{N_4}$, the following identities hold for the exponents of generators a and b in $\boldsymbol{S}$:*

$$a^x = a^{t+\iota\cdot u+v} = a^t a^{\iota\cdot u} a^v = a^t b^u a^v;\ a^y = a^{\iota\cdot t+u+\iota\cdot v} = a^{\iota\cdot t} a^u\ a^{\iota\cdot v} = b^t a^u b^v. \tag{21}$$

Identity to (21) can be extended for any word $w \in \boldsymbol{S}$:

$$w^x = w^{t+\iota\cdot u+v} = w^t \overline{w}^u w^v; w^y = w^{\iota\cdot t+u+\iota\cdot v} = \overline{w}^t w^u \overline{w}^v, \tag{22}$$

where the word $\overline{w}$ is obtained from w by renaming the generator a to b and b to a respectively. It is easily verified that the exponent function in $\boldsymbol{S}$ satisfies the following more general identities for any $w, w_1, w_2 \in \boldsymbol{S}$ and any $x, y \in \boldsymbol{NSR}$:

$$w^x w^y = w^{(x+y)} = w^{x+y}; \qquad (w^x)^y = w^{(x\cdot y)} = w^{x\cdot y}, (w_1 w_2)^x = (w_1)^x (w_2)^x. \tag{23}$$

The partial case of (23) are the following identities: $a^x\ a^y = a^{x+y}$, $(a^x)^y = a^{(x\cdot y)} = a^{x\cdot y}$ and $(a_1 a_2)^x = (a_1)^x (a_2)^x$ for any $a, a_{1,}\ a_2 \in \boldsymbol{S}$ and $x, y \in \boldsymbol{NSR}$. The same is valid for the generator b. The illustration of the computation of exponents in $\boldsymbol{S}$ is presented in Example 1 below.

The set of matrices defined over the $\boldsymbol{NSR}$ is denoted by $\boldsymbol{M_{NSR}}$.

Referencing to the Proposition 1, we present the following easily verifiable theorem without a proof.

Theorem 2. *$\boldsymbol{M_S}$ is a multiplicative $\boldsymbol{M_{NSR}}$-semibimodule.*

Since $\boldsymbol{NSR}$ is acting on the semigroup $\boldsymbol{S}$ as an exponent function, then $\boldsymbol{M_{NSR}}$ is acting on $\boldsymbol{M_S}$ as MPF. According to Definition 7 and semigroup $\boldsymbol{S}$ presentation in (12) the following proposition can be formulated.

Proposition 5. *$\boldsymbol{NSR}$ introduced in the Definitions 6 and 7 has non-commuting additive monoid <$\boldsymbol{NSR}$; +; 0> and commuting multiplicative monoid <$\boldsymbol{NSR}$; ·; 1>, i.e., for all $x, y, z_1, z_2 \in \boldsymbol{NSR}$ the following identities hold:*

$$z_1 + x + y + z_2 = z_1 + y + x + z_2. \tag{24}$$

$$x\cdot y = y\cdot x. \tag{25}$$

Relation (25) implies the following identity:

$$\iota\cdot x = x\cdot\iota. \tag{26}$$

Example 1. *The computation of w^x. Let $w = b^3aba^2$ and $x = 2 + \iota\cdot 3 + 4$, then the 1-st step of the computation is performed in the following way.*

$$w^x = w^{2+\iota\cdot 3\cdot+\ 4} = (b^3aba^2)^{2+\iota\cdot 3+4} = (b^3aba^2)^2\ (b^3aba^2)^{\iota\cdot 3}\ (b^3aba^2)^4 = (b^6a^2b^2a^4)\ (a^9b^3a^3b^6)\ (b^{12}a^4b^4a^{48})$$

At the second step transformation to the normal form (15), every word in parentheses is performed.

$$(b^6a^2b^2a^4)\ (a^9b^3a^3b^6)\ (b^{12}a^4b^4a^{48}) = (b^7aba^5)\ (a^{12}b^9)\ (b^{15}aba^{11}) = (b^3aba)\ (a^4b)\ (b^3aba^3) = = ba^9b^8a = bab^4a.$$

The final word is found using R_1 and R_2 as well, where $b^8 = b^4$, $a^9 = a$.

The gray part of Table 1 represents the additive subgroup $\boldsymbol{N_4}^+ = \{1, 2, 3, 4\}$ with neutral element equal to 4. Subgroup $\boldsymbol{N_4}^+$ has the subset of two generators $\boldsymbol{\Gamma} = \{1, 3\}$. The subgroup $\boldsymbol{N_4}^+$ and generators $\boldsymbol{\Gamma}$ will play an important role in proving the uniform distribution of conversations and the security against eavesdropping attack of MPF SIP in Section 5. Moreover, relations R_1, R_2 in (11) define the smallest exponent $e = 5$ where subgroup $\boldsymbol{N_4}^+$ has at least two generators. It is important to have a random choice in the set having at least two generators and will be used in our construction as well. The gray part of Table 2 represents the multiplicative semigroup $\boldsymbol{N_4}^+$ in $\boldsymbol{N}_4$. It is easily verified that $\boldsymbol{N_4}^+$ is a subsemiring of $\boldsymbol{N}_4$. Replacing $\boldsymbol{N}_4$ by $\boldsymbol{N_4}^+$ in Definition 7, we obtain a new near-semiring as a sub-semiring of $\boldsymbol{NSR}$ which is denoted by $\boldsymbol{sNSR}$. The matrix set over the $\boldsymbol{sNSR}$ is denoted by $\boldsymbol{M_{aNSR}}$ and matrix set with entries in $\boldsymbol{\Gamma} = \{1, 3\}$ is denoted by $\boldsymbol{M_\Gamma}$. MPFs defined by the multiplicative $\boldsymbol{M_{sNSR}}$-semibimodule $\boldsymbol{M_S}$ and by the multiplicative $\boldsymbol{M_{sNSR}}$-semibimodule $\boldsymbol{M_S}$ are denoted by MPF_S^{NSR} and MPF_S^{sNSR} respectively.

Together with the introduced here near-semirings the introduction of anti-$\boldsymbol{NSR}$, denoted by $\boldsymbol{aNSR}$, to the original $\boldsymbol{NSR}$ is necessary. According to (20), $\boldsymbol{aNSR}$ is defined by:

$$\boldsymbol{aNSR} = -\boldsymbol{N_4} - \iota\cdot\boldsymbol{N_4} - \boldsymbol{N_4} \cup -\iota\cdot\boldsymbol{N_4} - \boldsymbol{N_4} - \iota\cdot\boldsymbol{N_4}, \tag{27}$$

where for any element x in $\boldsymbol{NSR}$ there exists a unique element x' in $\boldsymbol{aNSR}$ obtained by switching the sign of x from positive to negative, i.e., $x' = -x$. The matrix set over the $\boldsymbol{aNSR}$ is denoted by $\boldsymbol{M_{aNSR}}$. For any matrix H in $\boldsymbol{M_{NSR}}$ there exists a unique matrix $-H$ in $\boldsymbol{M_{aNSR}}$ with negative entries. Then, formally, we assume that

$$H - H = O, \tag{28}$$

where O is the zero matrix, i.e., matrix with all entries equal to zero.

This construction will be used in the security proofs for the so-called simulator Sim computations. Several additional properties of MPF_S^{aNSR} are presented below without the proof. Let O be a zero matrix in $\boldsymbol{M_{NSR}}$ and E is a unity matrix in $\boldsymbol{M_S}$ consisting of all entries equal to 1 in $\boldsymbol{S}$. Then according to (7), (27) and (28), for any $W, A \in \boldsymbol{M_S}$ and $U, H \in \boldsymbol{M_{NSR}}$ the following identities hold:

$$A^O = E;\ {}^{O}A = E;\ W * E = E * W = W; \tag{29}$$

$${}^{U}A^{H} * {}^{U}A^{-H} = {}^{U}(A^{H-H}) = {}^{U}A^{O} = {}^{U}E = E. \tag{30}$$

The last identity remains valid if the presented actions are reversed from the left to the right.

4. MPF Sigma Identification Protocol (SIP)

In general, sigma identification protocols (SIP) are realized using the conversation between the Prover and the Verifier when the Prover proves to the Verifier the knowledge of the secret (e.g., his private key—witness) without revealing knowledge about this secret [19]. In this case it is said that SIP has the Zero Knowledge Proof (ZKP) property [19]. Prover is using his private, public key pair we denote by PrK, PuK named as a witness-statement pair.

We denote any matrix Q that is generated uniformly at random from the matrix set $\boldsymbol{M}$ by $Q \leftarrow \mathrm{rand}(\boldsymbol{M})$.

We use matrix sets $\boldsymbol{M_{sNSR}}$ and $\boldsymbol{M_\Gamma}$ introduced in Section 3 instead of the matrix set $\boldsymbol{M_{NSR}}$ to provide a random uniform distribution of data generated in MPF SIP.

Parties share the same public parameter represented by matrix W in $\boldsymbol{M_S}$ generated at random $W \leftarrow \mathrm{rand}(\boldsymbol{M_S})$. The prover runs the following key pair generation algorithm. For the private key PrK-witness generation two secret matrices X, Y in $\boldsymbol{M_\Gamma} = \{1, 3\}$ are chosen at random:

$$X{,}Y \leftarrow \mathrm{rand}(\boldsymbol{M_\Gamma}). \tag{31}$$

Then PrK = $(X, Y) \in \boldsymbol{M_\Gamma} \times \boldsymbol{M_\Gamma}$.

The public key PuK-statement is computed using MPF_S^{sNSR} defined above:

$$\mathrm{PuK} = {}^{X}W^{Y} = A. \tag{32}$$

Prover distributes his PuK = $A \in \boldsymbol{M_S}$ to all users including Verifier. According to the 3-rd condition in the Proposition 2 represented by the distributive identity (7), we formulate the following easily verified theorem without proof.

Theorem 3. *MPF_S^{sNSR} satisfies the distributive identity (7).*

Definition 8. *Equation (32) defines a set of relations **Rel** between the set of witnesses (PrK) and statements (PuK) which is a subset of the following direct product of sets:* $\mathbf{Rel} \subseteq (\mathbf{M_\Gamma} \times \mathbf{M_\Gamma}) \times \mathbf{M_S}$.

Since relations R_1, R_2 in (11) define a finite semiring $\boldsymbol{S}$, they induce the finiteness of $s\boldsymbol{NSR}$. Then, sets $\boldsymbol{M_{sNSR}}$ and $\boldsymbol{M_S}$ are finite as well.

Definition 9. *Let matrix $A = \{a_{ij}\}$ be of finite order and assume that all entries q_{ij} can be effectively encoded by the finite string of bits not exceeding polynomial length. Then, matrix A is effectively recognizable in $\boldsymbol{M_S}$ if all its entries can be effectively decoded and effectively transformed to the normal form (15).*

Proposition 6. *Any finite length word w in **S** can be transformed to the normal form (15) using the linear number of operations with respect to the length of w.*

Definition 10. *Relation **Rel** is efficiently recognizable if every matrix $A' = \{a_{ij}'\}$, where $a_{ij}'$are finite strings of generators a, b in $\boldsymbol{M_S}$, can be effectively transformed into the matrix A in $\boldsymbol{M_S}$ with all entries expressed in the normal form (15).*

Definition 11. *Relation is an effective relation if it is efficiently recognizable.*

Proposition 7. *Relation **Rel** is effective.*

Referencing to the general definition of Sigma protocol in [19], the corresponding definition can be formulated for MPF SIP.

Definition 12. *Let **Rel** $\subseteq (\mathbf{M_\Gamma} \times \mathbf{M_\Gamma}) \times \boldsymbol{M_S}$ be an effective relation. An MPF SIP for **Rel** is a pair (P, V) of interactive protocols executed by the Prover and the Verifier. Protocol P is taking a witness-statement pair (PrK, PuK) $\in$ **Rel** as an input. Protocol V is taking as an input statement PuK $\in \boldsymbol{M_S}$. Then after the conversation V outputs **accept** or **reject**.*

MPF SIP is performed during three pass communications named as a conversation between the Prover and the Verifier.

1. Prover generates two matrices U, $V \leftarrow \text{rand}(\boldsymbol{M_\Gamma})$ at random and using his witness-PrK computes the **commitment** $C = (C_0, C_1, C_2)$ consisting of three matrices C_0, C_1, C_2 in $\boldsymbol{M_S}$:
$$C_0 = {}^{U}W^{V},\ C_1 = {}^{U}W^{Y},\ C_2 = {}^{X}W^{V}. \tag{33}$$
Prover sends C to the Verifier.
2. After receiving C, Verifier generates two matrices H', $H'' \leftarrow \text{rand}(\boldsymbol{M_{sNSR}})$ at random and independently, forms **challenge** $H = (H', H'')$ and sends H to the Prover.
3. Upon receiving H, Prover computes the **response** $R = (S, T)$ consisting of two matrices S, T in $\boldsymbol{M_{sNSR}}$:
$$S = U + H'X,\ T = V + YH'', \tag{34}$$
and sends $R = (S, T)$ to the Verifier.

At this stage Prover and Verifier complete the conversation. After receiving $R = (S, T)$, Verifier checks if
$$ {}^{S}W^{T} = C_0 * C_1{}^{H''} * {}^{H'}C_2 * {}^{H'}A^{H''}, \tag{35}$$
and if it is the case outputs ***accept***.

The distinct feature of the proposed protocol (against, e.g., Schnorr or Okamoto protocols, [19]) is that the Prover generates a commitment at the first step of the protocol using components X, Y of the witness-PrK = (X, Y).

Completeness. On the common statement input PuK = A, the honest Prover knows witness-PrK = (X, Y) for PuK and succeeds in convincing the Verifier of his knowledge with probability 1. It follows from the validity of associativity identity (6) and distribution identity (7):

$$^{S}W^{T} = {}^{(U+H'X)}W^{(V+YH'')} = {}^{U}W^{V} * {}^{U}W^{YH''} * {}^{H'X}W^{V} * {}^{HX'}W^{YH'} = {}^{U}W^{V} * ({}^{U}W^{Y})^{H''} * {}^{H'}({}^{X}W^{V}) * {}^{H'}({}^{X}W^{Y})^{H'} = C_0 * C_1{}^{H''} * {}^{H'}C_2 * {}^{H'}A^{H''}.$$

Verifier uses the conversation (C, H, R) together with the Prover's statement PuK = A for the verification and yields accept if (35) holds.

The test example of this protocol is presented in Appendix A.

5. Security Analysis

We consider the main three main kind of attacks for SIP presented in [19] ordered by their power, i.e., direct attack, eavesdropping attack and active attack. The weakest attack of the three is a direct attack and it is applied mainly for password protected systems which can also be realized using symmetric cryptography. The outcome of these attacks is either adversary impersonation of legal Prover or even compromisation of legal prover's secret, namely his password or private key PrK-witnwss. The detailed description of the attack game and the theorem formulating security against this attack is presented in ([19], Section 18.3). Since this attack is not of direct interest for our research, we only use the security formulation for this attack in our construction in Theorem 4 below as an intermediate result to consider the more powerful eavesdropping attack. In this section we prove the conditions under which the proposed MPF SIP is resistant against eavesdropping attack finalizing it in Theorems 5 and 6. Unfortunately, we were not able to prove the resistance against most powerful active attack for the reasons presented below.

Assumption 1. *MPF_S^{NSR} is a candidate for one-way function (OWF).*

This assumption can be supported by our previous results presented in [11–14]. The MPF is constructed using similar algebraic structures as in [14]. In [12,13] the NP-completeness of the similar MPF problem is proven. MPF function introduced there is defined in finite modified medial platform semigroup and finite power near-semiring. Despite the lack of proof of the NP-completeness of the MPF problem defined here, we can present some links of this MPF with the well-known multivariate quadratic (MQ) problem which is proved to be NP-complete over any field [7]. Let φ: $\boldsymbol{S} \rightarrow \boldsymbol{S_a}$ is the homomorphism of the semigroup $\boldsymbol{S}$ to the semigroup $\boldsymbol{S_a}$ defined by the introduced new relation b = 1. Then $\boldsymbol{S_a}$ = {1, a, a^2, a^2, a^4} is a cyclic monoid with index 1 and period 4 [20]. The corresponding public matrix denoted by W'. Then the entries of this public matrix W' consist of elements a^i, where $I \in$ {0, 1, 2, 3, 4} due to relation R_1. Analogously, the ***NSR*** can be homomorphically transformed to the set of integers $\boldsymbol{Z}_5$ = {0, 1, 2, 3, 4} by introducing the relation ι = 0. Then the matrices X', $Y' \leftarrow$ rand($\boldsymbol{Z}_5$) are generated.

Since $\boldsymbol{S_a}$ is a cyclic semigroup then the discrete logarithm operation dlog$_a$ with the base a can be applied elementwise to the MPF relation $^{X'}(W')^{Y'} = A'$. In this case we obtain MQ problem but defined not over the field $\boldsymbol{F}_5$ = {0, 1, 2, 3, 4} (where operations are defined mod 5) since according to relation R_1 in (11) the exponents of generator a cannot be reduced mod 5. It seems that the obtained MQ problem represented by matrix equation

$$\mathrm{dlog}_a({}^{X'}(W')^{Y'}) = \mathrm{dlog}_a(A'),$$

is at least no less complex than the "standard" MQ problem. Then, we can make an assumption that MPF introduced in (4), (32), where unknown monomials are in exponents is a candidate for one-way function (OWF).

This assumption is required to prove that the proposed MPF SIP is secure against the eavesdropping attack and to select the values of the security parameters.

Referencing to Assumption 1 and [19], we can formulate the following theorem.

Theorem 4. *If MPF_S^{NSR} is a candidate for OWF and the challenge space is super-poly, then MPF SIP identification protocol is secure against the direct attack.*

Proof. According to the Assumption 1, MPF_S^{NSR} is a candidate for OWF.

Referencing to the relations R_1, R_2 in (11), the challenge space is exponential with respect to the order m of matrices in $\boldsymbol{M_{NSR}}$. The cardinality of $\boldsymbol{M_{NSR}}$ is no less than $5^{2\text{mm}}$, thus, it is super poly as well. □

It is easily verified that Assumption 1 and Theorem 4 can be applied to MPF_S^{sNSR}.

In order to formulate the security against the eavesdropping attack, we need the definition and the proof of the Honest Verifier Zero Knowledge (HVZK) property [19] for MPF SIP. We realized that instead of HVZK of MPF SIP we can prove a rather stronger result denoted as a special HVZK.

Definition 13. *Identification protocol is a special HVZK if there exists an efficient probabilistic algorithm called a simulator Sim such that for all possible witness-statement pairs or private and public key pairs (PrK, PuK) the following two conditions hold: 1) the output distribution of Sim on input (PuK, H) is identical to the distribution of transcript of a conversation (C, H, R) between Prover on input (PrK, PuK) and Verifier on input Puk, and 2) for all inputs PuK and H, algorithm Sim always outputs a pair (C, R) such that (C, H, R) is an accepting conversation for PuK.*

Following the methodology presented in [19], the following theorem can be formulated.

Theorem 5. *MPF SIP protocol is a special HVZK.*

Proof. On Sim input R', H' and PuK, the valid commitment $C' = (C_{0'}, C_{1'}, C_{2'})$ and corresponding transcript of a conversation (C', H', R') must be generated with identical distribution as (C, H, R) between the Prover and the Verifier. □

Lemma 2. *Let W be a public parameter and two pairs of matrices X,Y ← rand($\boldsymbol{M_\Gamma}$), X′,Y′ ← rand($\boldsymbol{M_\Gamma}$) are generated uniformly at random, then the entries of matrices $A = {}^{X}W^{Y}$ and $A' = {}^{X'}W^{Y'}$ computed according to (32) are uniformly distributed.*

Proof of Lemma. Define two subsemigroups in $\boldsymbol{S}$, namely $\boldsymbol{S_a} = \{a, a^2, a^3, a^4\}$, $\boldsymbol{S_b} = \{b, b^2, b^3, b^4\}$ and a set of exponents of generators in $\boldsymbol{S}$ denoted earlier by $\boldsymbol{N_4}^+ = \{1, 2, 3, 4\}$. Then exponent function of generator a in $\boldsymbol{S_a}$ provides the following 1-to-1 mapping $\exp_a$: $\boldsymbol{N_4}^+ \to \boldsymbol{S_a}$. Since $\exp_a$ is 1-to-1, then for any $I \leftarrow \text{rand}(\boldsymbol{N_4}^+)$ the value $\exp_a(i) = a^i$ will have a uniform random distribution. The same is valid for the function $\exp_b$: $\boldsymbol{N_4}^+ \to \boldsymbol{S_b}$. Let $\exp_{a,a}$: $\boldsymbol{N_4}^+ \times \boldsymbol{N_4}^+ \to \boldsymbol{S_a}$ is a function defining multiplication of generator a^i by generator a^j, where $i, j \leftarrow \text{rand}(\boldsymbol{N_4}^+)$, i.e., $\exp_{a,a}(i, j) = a^i a^j = a^{i+j}$. According to R_1 in (11), function $\exp_{a,a}(i, j)$ provides a 4-to-1 mapping and hence the value a^e obtained after the reduction of exponent a^{i+j} using R_1 will have a uniform random distribution.

Let us consider the double exponent function expexp_a: $\boldsymbol{N_4}^+ \times \Gamma \to \boldsymbol{S_a}$, where $\text{expexp}_a(i, k) = (a^i)^k$ with $I \in \boldsymbol{N_4}^+$, $k \in \Gamma$. This function provides a 2-to-1 mapping. Let i, k values are random and uniformly distributed, then the value $(a^i)^k$ is also random and uniformly distributed. The same is valid for the generator b and for complex exponents $\iota \cdot i$ and $\iota \cdot k$. In the latter case complex unit ι simply changes a to b and vice versa.

As a consequence, the mentioned above exponents of generators a, b are random and uniformly distributed.

If we have any word $w \in \boldsymbol{S}$ and exponentiate it by i, then after the reduction of exponents corresponding to the generators a and b the uniform distribution of resulting exponents will be obtained. After that the generators are grouped according to the normal form defined in (15) and this grouping simply corresponds to computing expressions of the type $a^i a^j = a^{i+j}$. Hence the grouping procedure will not change the uniform distribution. As a result, we obtain uniformly distributed normal form w_η in (14).

These results are applied subsequently to the left MPF and right MPF (according to Definitions 1 and 2) to prove the uniform distribution of entries of matrices A and A' being a value of two sided MPF in Definition 3. The proof is performed by induction with respect to the order m of MPF matrices. The first step of induction for $m = 1$ is proved above.

The Lemma is proved. □

Lemma 3. *Suppose the following pairs of matrices X, Y* $\leftarrow$ *rand(*$\boldsymbol{M}_{\boldsymbol{\Gamma}}$*), U, V* $\leftarrow$ *rand(*$\boldsymbol{M}_{\boldsymbol{\Gamma}}$*),* X'*,* Y' $\leftarrow$ *rand(*$\boldsymbol{M}_{\boldsymbol{\Gamma}}$*),* U'*,* V' $\leftarrow$ *rand(*$\boldsymbol{M}_{\boldsymbol{\Gamma}}$*) and* $H^{\wedge\prime}$*,* $H^{\wedge\prime\prime}$ $\leftarrow$ *rand(*$\boldsymbol{M}_{\boldsymbol{sNSR}}$*) are generated uniformly at random. Then the distribution of entries of matrices S, T in (34) and of matrices* S'*,* T' *computed by the relations.*

$$S' = U' + H^{\wedge\prime}X',\ T' = V' + Y'H^{\wedge\prime\prime}, \tag{36}$$

have the same uniform random distribution.

Proof of Lemma. Firstly, we prove that the product of matrices H′X, YH″ has the same uniform random distribution as the product of matrices H^′X′, Y′H^″. Let us consider a scalar case with $h'x$, yh'' and $h^{\wedge\prime}x'$, $y'h^{\wedge\prime\prime}$ instead, where h', h'', $h^{\wedge\prime}$, $h^{\wedge\prime\prime}$ $\in \boldsymbol{N}_4{}^+$ and x, y, x', y' $\in \boldsymbol{\Gamma}$. Then the multiplication function is defined by the mapping $\mathrm{mul}_{h,x}$: $\boldsymbol{N}_4{}^+ \times \boldsymbol{\Gamma} \rightarrow \boldsymbol{N}_4{}^+$ which according to Table 2 is 2-to-1. Since multipliers are chosen uniformly at random, then the value $\mathrm{mul}_{h,\,x}(h, x) = hx$ in $\boldsymbol{N}_4{}^+$ is distributed uniformly at random. The same is valid for other terms.

Now consider the addition function $\mathrm{add}_{u,z}$: $\boldsymbol{\Gamma} \times \boldsymbol{N}_4{}^+ \rightarrow \boldsymbol{N}_4{}^+$ which according to Table 1 is 2-to-1. Then, since the value $z = hx$ is distributed uniformly at random the value $\mathrm{add}_{u,\,z}(u, z) = s$ is also distributed uniformly at random.

The random and uniform distribution of entries of matrices $H'X$, YH'' and $H^{\wedge\prime}X'$, $Y'H^{\wedge\prime\prime}$ can be proved by induction using the uniform and random distribution of values of functions $\mathrm{mul}_{h,\,x}$ and $\mathrm{add}_{u,\,z}$. Then, the entries of matrices S, T and S', T' are distributed uniformly at random as well.

The Lemma is proved. □

Proof. Proceeding with the proof of the theorem, the response $R' = (S', T')$ must be computed referencing to (34). Then, the following random matrices in $\boldsymbol{M}_{\boldsymbol{NSR}}$ are generated independently: U', V' $\leftarrow$ rand($\boldsymbol{M}_{\boldsymbol{\Gamma}}$), X', Y' $\leftarrow$ rand($\boldsymbol{M}_{\boldsymbol{\Gamma}}$). As an additional input the simulator takes two challenge matrices $H^{\wedge\prime}$, $H^{\wedge\prime\prime}$ $\leftarrow$ rand($\boldsymbol{M}_{\boldsymbol{sNSR}} \times \boldsymbol{M}_{\boldsymbol{sNSR}}$). Then, according to (34):

$$S' = U' + H^{\wedge\prime}X',\ T' = V' + Y'H^{\wedge\prime\prime}. \tag{37}$$

Referencing to Lemmas 2 and 3, $R' = (S', T')$ and $R = (S, T)$ have the same uniform distribution and hence the distribution of ${}^{S'}W^{T'}$ is the same as the distribution of ${}^{S}W^{T}$ in (35). According to the Theorem 3 ${}^{S'}W^{T'}$ has the following expression

$${}^{S'}W^{T'} = {}^{(U'+H^{\wedge\prime}X')}W^{(V'+Y'H^{\wedge\prime\prime})} = {}^{U'}W^{V'} * {}^{U'}W^{Y'H^{\wedge\prime\prime}} * {}^{H^{\wedge\prime}X'}W^{V'} * {}^{H^{\wedge\prime}X'}W^{Y'H^{\wedge\prime\prime}} = {}^{U'}W^{V'} * ({}^{U'}W^{Y'})^{H^{\wedge\prime\prime}} * {}^{H^{\wedge\prime}}({}^{X'}W^{V'}) * {}^{H^{\wedge\prime}}B^{H^{\wedge\prime\prime}}, \tag{38}$$

where $B = {}^{X'}W^{Y'}$.

For given matrices U', V', X', Y' and $(H^{\wedge\prime}, H^{\wedge\prime\prime})$ generated uniformly at random and independently, Sim must compute a challenge $C' = (C_{0'}, C_{1'}, C_{2'})$ satisfying the following equation:

$${}^{S'}W^{T'} = C_{0'} * (C_{1'})^{H^{\wedge\prime\prime}} * {}^{H^{\wedge\prime}}(C_{2'}) * {}^{H^{\wedge\prime}}A^{H^{\wedge\prime\prime}} \tag{39}$$

Referencing to identity (37), Sim computes

$$C_{0'} = {}^{U'}W^{V'},\ C_{1'} = {}^{U'}W^{Y'},\ C_{2'} = {}^{X'}W^{V'} * B^{H^{\wedge\prime\prime}} * A^{-H^{\wedge\prime\prime}}. \tag{40}$$

According to Theorem 5, the entries of matrices $C_{0'}$, $C_{1'}$, have the same uniform distribution as of matrices C_0, C_1. Since (1) the distribution of ${}^{S'}W^{T'}$ is the same distribution as of ${}^{S}W^{T}$, (2) ${}^{X'}W^{V'}$ is uniformly distributed and (3) the computation of matrix $C_{2'}$ in (39) is based on Hadamar multiplication rule [16] (i.e., elementwise), then the distribution of $C_{2'}$ is the same as the distribution of C_2. Then the commitment $C' = (C_{0'}, C_{1'}, C_{2'})$ has the same distribution as $C = (C_0, C_1, C_2)$. It remains to prove that (C', H', R') is an accepting conversation using the following identities.

$$\begin{aligned} C_{0'} * (C_{1'})^{H\hat{}'' * H\hat{}'}(C_{2'})^{* H\hat{}'} A^{H\hat{}''} &= {}^{U'}W^{V'} * ({}^{U'}W^{Y'})^{H\hat{}' * H\hat{}''}({}^{X'}W^{V'})^{* H\hat{}'} B^{H\hat{}'' * H\hat{}'} A^{-H\hat{}'' * H\hat{}'} A^{H\hat{}''} \\ &= {}^{S'}W^{T' * H\hat{}'} A^{H\hat{}''\text{-}H\hat{}''} = {}^{S'}W^{T' * H\hat{}'} A^{O} = {}^{S'}W^{T'} * E = {}^{S'}W^{T'}. \end{aligned} \tag{41}$$

The theorem is proved. □

Since special HVZK implies HVZK, then according to the Assumption 1 and Theorems 4 and 5, proposed MPF SIP is secure against the direct attack and is HVZK. The Theorem 19.3 in [19] states that if an identification protocol is secure against direct attacks, and is HVZK, then it is secure against eavesdropping attacks. Referencing to this result we have proven the following theorem.

Theorem 6. *MPF SIP is secure against the eavesdropping attack.*

Unfortunately, we are not able to prove the security of MPF SIP against an active attack since we have not proved the soundness of MPF SIP. Our construction is based on far more complicated algebraic structures than many traditional identification protocols including Schnorr protocol.

6. Selection of the Security Parameters and Efficiency Analysis

Since relations R_1, R_2 are fixed, the security parameter is the order m of matrices defining MPF. Then, according to (32) PrK and PuK matrix relation in the Definition 8 consists of m^2 exponent equations of the type (4). According to the Assumption 1 and the belief that the solution of randomly generated MQ system is hopeless when system consists of $n \geq 80$ equations with $v \geq 80$ variables [9], it is sensible to choose the number of exponent equations of the type (4) corresponding to the matrix equation (32) to be no less than 80. Then, m can be chosen to be equal to 10, 11, 12. In this case the number of exponent equations is equal to 100, 121, 144 correspondingly. To represent ***NSR*** elements of the types $t + \iota{\cdot}u + v$ and $t{\cdot}\iota + u + v{\cdot}\iota$, where $t, u, v \in \{1, 2, 3, 4\}$ 9 bits are required. Thus, memory requirement for PrK = (X, Y) is $2 \times 9\ m^2$ bits. To represent the word w in ***S*** in normal form (15) 6 bits are required. So, PuK = A representation requires $6\ m^2$ bits.

Effectivity of SIP is related to the left and right MPF value computations in (2) and (3) and can be performed using exponentiation tables of the size 4x4 in our case. Transformation of matrix entries to the normal form requires asymptotically O (m^2) operations. The computational resources for the one-sided MPF value computation are equivalent to the matrix multiplication and are asymptotically at most O (m^3). SIP realization for Prover requires 6 one-sided MPF values computation, 2 multiplication of matrices in ***NSR*** requiring O (m^3) operations and two additions of matrices in ***NSR*** requiring O (m^2) operations. Both matrix multiplication and addition can be performed using the table of operations of the size 4×4 as shown in Tables 1 and 2.

For the Verifier's side one must compute MPF values presented in (35). It takes two one-sided MPF computations for the left side of (35) and six one-sided MPF computations. Hence asymptotically it takes O (m^3) operations.

7. Discussion and Conclusions

It was an intriguing idea for the authors to create and analyze Sigma identification protocol (SIP) based on the matrix power function (MPF) defined over the specially selected algebraic structures, namely modified platform medial semigroup ***S*** and power near-semiring (***NSR***). The initial medial semigroup is infinite, cancellative, multiplicative and non-commuting and is chosen to have two generators a and b. The modification of this medial semigroup is performed by introducing two extra relations R_1 and R_2 for the generators. It induces certain homomorphism yielding finite semigroup and preserving other properties of medial semigroup. In order to construct MPF, the certain power ***NSR*** is constructed. The properties of ***NSR*** are induced by the generic relation R_M of medial semigroup which makes addition operation non-commuting. Therefore, the notion of ***NSR*** is applied. The reason is that constructed ***NSR*** is simply not a semiring.

These algebraic structures are far more complicated than the structures used currently in well-known identification and sigma identification protocols, e.g., Schnorr or Okamoto protocols. They are based on relatively simple algebraic structures, namely cyclic groups ${\boldsymbol{Z}_{\boldsymbol{p}}}^{*}$ or $\boldsymbol{G}_{\boldsymbol{q}}$. MPF SIP construction based on more complicated algebraic structures was successful since a very important property of MPF presented in Proposition 2 was satisfied. In this connection we are expecting that proposed MPF SIP should provide greater security than existing Sigma protocols based on numerical cyclic groups. The investigation of the resistance against quantum cryptanalysis attack is very attractive and could be dedicated for the future research.

MPF, presented here, has some similarity to the certain MPF problem which was proven to be NP-complete in our previous publications [12,13]. It is believed so far that NP-complete problems cannot be effectively solved by quantum computers. The security of MPF SIP presented here relies on the assumption that this MPF problem is a candidate for one-way function (OWF).

Following the methodology, notions and security analysis of Sigma protocols presented in D. Boneh, and V. Shoup tutorial we proved that the proposed SIP is secure against the eavesdropping attack. Unfortunately, the proof that this protocol is secure against active attack was not presented since we have not proved the soundness of this protocol yet. The soundness of identification protocols is easily proven for simple algebraic structures, namely ${\boldsymbol{Z}_{\boldsymbol{p}}}^{*}$ or $\boldsymbol{G}_{\boldsymbol{q}}$. In our case, however, the algebraic structures are far more complicated and existing proof methodology used in cyclic groups cannot be applied.

Author Contributions: Conceptualization, methodology and investigation E.S.; Software, validation, writing—review and editing I.T.; Software, writing—original draft preparation A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: Many thanks to the Reviewers and their remarks, allowing us to substantially improve the security proof as well as the overall quality of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Symbol	**Explanation**
ι	Imaginary unit in ***NSR***, wiyh the property $\iota^2 = 1$.
A	Public key (PuK) matrix and statement in MPF SIP.
a, b	Generators in modified medial semigroup ***S***.
aNSR	Anti Near-semiring to the ***NSR***.
$C = (C_0, C_1, C_2)$	Commitment consisting of three matrices C_0, C_1, C_2 and computed by the Prover.
DSA	Digital signature algorithm
ECDSA	Elliptic curve digital signature algorithm
H', H''	Challenge matrices generated by Verifier.
HFE	Hidden field equations cryptosystem

Symbol	Explanation
$\boldsymbol{M_{NSR}}$	Set of matrices over ***NSR***.
MPF	Matrix Power Function.
$MPF_S{}^{NSR}$	Matrix Power Function defined by the matrices over base semiring ***S*** and exponent matrices over ***NSR***.
$\boldsymbol{M_S}$	Set of matrices defined over semiring ***S***.
$\boldsymbol{N_0}$	Semiring of natural numbers with zero.
NSR	Near-semiring.
OWF	One-way function
PrK = (X, Y)	Private key or witness in MPF SIP consisting of two generated at random matrices X, Y over the ***NSR***.
PuK = A	Public key and statement for PrK consisting of matrix A over the semiring ***S***.
Rel	Effective relation for MPF SIP relating a witness-PrK with statement PuK.
RSA	Rivest, Shamir, Adleman cryptosystem
S	Modified medial semigroup.
s, t, u, v	Positive integers.
SIP	Sigma Identification Protocol.
$\boldsymbol{S_M}$	Medial semigroup.
sNSR	Sub-near-semiring in ***NSR***.
V, U	Random matrices generated in ***NSR*** for commitment C computation.
w	Word in medial semigroup $\boldsymbol{S_M}$ or modified medial semigroup ***S***.
X, Y	Component matrices of PrK and witness over the ***NSR***.
x, y, z	Exponents of generators in ***NSR***.

Appendix A. Numerical Example of the MPF SIP

A numerical illustration of the MPF SIP is presented below for the matrices with dimensions 3 × 3. Firstly, PrK = (X, Y) components are generated, where X,Y ← rand($\boldsymbol{M_{NSR}}$). The entries of X, Y are chosen in the form $\{t + \iota \cdot u + v\}$ as elements in ***NSR*** (see Proposition 4), where t, u, v are in $\boldsymbol{N_4}^+$ = {1, 2, 3, 4} according to Corollary 5.6. For convenience the imaginary unit ι in (17) is replaced by latin notation i.

$$X = \begin{bmatrix} 1+i+1 & 3+3i+3 & 1+3i+1 \\ 1+3i+1 & 1+i+1 & 3+3i+3 \\ 3+3i+3 & 1+3i+1 & 1+i+1 \end{bmatrix}$$

$$Y = \begin{bmatrix} 3+i+3 & 3+3i+1 & 1+i+3 \\ 1+i+3 & 3+i+3 & 3+3i+1 \\ 3+3i+1 & 1+i+3 & 3+i+3 \end{bmatrix}$$

Next, Public matrix W is generated at random (W ← rand($\boldsymbol{M_S}$)) and PuK = A= ${}^XW^Y$ is computed referencing to (4), (21)–(23), (32).

$$W = \begin{bmatrix} ba^3b^3a & ba^2ba & bab^3a \\ bab^3a & ba^2b^3a & bab^2a \\ bab^3a & bab^3a & ba^3b^3a \end{bmatrix}$$

$$A = \begin{bmatrix} ba^4 & ba^3b^2a & bab^2a \\ ba^3b^2a & ba^4 & ba^2 \\ ba^2 & bab^2a & ba^3b^2a \end{bmatrix}$$

Two matrices U, V are generated U,V ← rand($\boldsymbol{M_{sNSR}}$) for the computation of **commitment**.

$$U = \begin{bmatrix} 1+3i+3 & 3+i+3 & 3+3i+1 \\ 3+3i+1 & 1+3i+3 & 3+i+3 \\ 3+i+3 & 3+3i+1 & 1+3i+3 \end{bmatrix}$$

$$V = \begin{bmatrix} 3+i+1 & 3+3i+3 & 1+1i+3 \\ 1+1i+3 & 3+i+1 & 3+3i+3 \\ 3+3i+3 & 1+1i+3 & 3+i+1 \end{bmatrix}$$

Then, according to (33), Prover computes the **commitment** $C = (C_0, C_1, C_2)$ consisting of three matrices C_0, C_1, C_2 in $\boldsymbol{M_S}$ and sends it to the Verifier.

$$C_0 = \begin{bmatrix} ba^2 & ba^3b^2a & ba^3b^2a \\ ba^2 & ba^3b^2a & ba^3b^2a \\ ba^3b^2a & ba^2 & ba^2 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} ba^4 & ba^3b^2a & bab^2a \\ ba^4 & ba^3b^2a & bab^2a \\ bab^2a & ba^2 & ba^4 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} ba^2 & ba^3b^2a & ba^3b^2a \\ bab^2a & ba^4 & ba^4 \\ ba^4 & bab^2a & bab^2a \end{bmatrix}$$

Verifier generates the **challenge** consisting of two matrices $H', H'' \leftarrow \text{rand}(\boldsymbol{M_{sNSR}})$:

$$H' = \begin{bmatrix} 1+i+1 & 2+2i+2 & 3+3i+3 \\ 3+3i+3 & 1+i+1 & 2+2i+2 \\ 2+2i+2 & 3+3i+3 & 1+i+1 \end{bmatrix}$$

$$H'' = \begin{bmatrix} 4+4i+4 & 2+3i+4 & 1+2i+4 \\ 1+2i+4 & 4+4i+4 & 2+3i+4 \\ 2+3i+4 & 1+2i+4 & 4+4i+4 \end{bmatrix}$$

Upon receiving H', H'' the Prover computes the response matrices S, T according to (34) where $S = U + H'X$, $T = V + YH''$. The entries of these matrices are the exponents of generators a, b in semigroup $\boldsymbol{S}$ and are presented in non-reduced form.

$$S = U + H'X = \begin{bmatrix} 1+59i+67 & 1+45i+51 & 1+47i+55 \\ 1+47i+55 & 1+59i+67 & 1+45i+51 \\ 1+45i+51 & 1+47i+55 & 1+59i+67 \end{bmatrix}$$

$$T = V + YH'' = \begin{bmatrix} 1+74i+108 & 1+80i+110 & 1+72i+104 \\ 1+72i+104 & 1+74i+108 & 1+80i+110 \\ 1+80i+110 & 1+72i+104 & 1+74i+108 \end{bmatrix}$$

After reduction using relations R_1, R_2 in (11) the Prover computes the **response** $R = (S, T)$ and sends it to the Verifier, where matrices S, T are expressed in the following way:

$$S = U + H'X = \begin{bmatrix} 1+3i+3 & 1+i+3 & 1+3i+3 \\ 1+3i+3 & 1+3i+3 & 1+i+3 \\ 1+i+3 & 1+3i+3 & 1+3i+3 \end{bmatrix}$$

$$T = V + YH'' = \begin{bmatrix} 1+2i+4 & 1+4i+2 & 1+4i+4 \\ 1+4i+4 & 1+2i+4 & 1+4i+2 \\ 1+4i+2 & 1+4i+4 & 1+2i+4 \end{bmatrix}$$

Upon receiving $R = (S, T)$, Verifier checks if the identity (35) holds:

$${}^{S}W^{T} = C_0\, C_1^{H''}\; {}^{H'}C_2\; {}^{H'}A^{H''} = \begin{bmatrix} ba^2ba & b^2a & b^4a \\ b^4a & ba^2b^3a & ba^2ba \\ b^2a & ba^2ba & ba^2b^3a \end{bmatrix}$$

Since in this case the identity (35) is satisfied, the Verifier outputs **accept**.

References

1. Myasnikov, A.; Shpilrain, V.; Ushakov, A. *Group-Based Cryptography*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
2. Myasnikov, A.G.; Shpilrain, V.; Ushakov, A. *Non-Commutative Cryptography and Complexity of Group-Theoretic Problems*; American Mathematical Society: Providence, RI, USA, 2011.
3. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509. [CrossRef]
4. Garey, M.; Johnson, D. *Computers and Intractability: A Guide to Theory of NP-Completeness*; H. Freeman: New York, NY, USA, 1979.
5. Chen, L.; Jordan, S.; Liu, Y.; Moody, D.; Peralta, R.; Perlner, R.; Smith-Tone, D. *Report on Post-Quantum Cryptography*; US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2016; Volume 12.
6. Micciancio, D.; Regev, O. Lattice-based cryptography. In *Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 147–191.
7. Patarin, J.; Goubin, L. Trapdoor One-Way Permutations and Multivariate Polynomials. In Proceedings of the First International Conference on Information and Communication Security, LNCS, Beijing, China, 11–14 November 1997; Volume 1334, pp. 356–368.
8. Wolf, C. Hidden Field Equations. (HFE)-Variations and Attacks. Ph.D. Thesis, Ruhr-University, Bochum, Germany, 2002.
9. Faugere, J.; Antoine, J. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Grobner bases. In *Advances in Cryptology-CRYPTO*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 44–60.
10. Yasuda, T.; Dahan, X.; Huang, Y.-J.; Takagi, T.; Sakurai, K. MQ Challenge: Hardness Evaluation of Solving Multivariate Quadratic Problems. Available online: http://eprint.iacr.org/2015/275.pdf (accessed on 1 July 2021).
11. Sakalauskas, E. The Multivariate Quadratic Power Problem over Zn is NP-complete. *Inf. Technol. Control* **2012**, *41*, 33–39. [CrossRef]
12. Sakalauskas, E.; Mihalkovich, A. MPF Problem over Modified Medial Semigroup Is NP-complete. *Symmetry* **2018**, *10*, 571. [CrossRef]
13. Mihalkovich, A.; Sakalauskas, E.; Luksys, K. Key Exchange Protocol Defined over a Non-Commuting Group Based on an NP-complete Decisional Problem. *Symmetry* **2020**, *12*, 1389. [CrossRef]
14. Sakalauskas, E. Enhanced matrix power function for cryptographic primitive construction. *Symmetry* **2018**, *10*, 43. [CrossRef]
15. Inassaridze, N.; Kandelaki, T.; Ladra, M. Categorical interpretations of some key agreement protocols. *J. Math. Sci.* **2013**, *195*, 439–444. [CrossRef]
16. Horn, R.A.; Johnson, C.R. *Matrix Analysis*; Cambridge University Press: Cambridge, UK, 2012.
17. Chrislock, J.L. On medial semigroups. *J. Algebra* **1969**, *12*, 1–9. [CrossRef]
18. Krishna, K.V. Near-Semirings: Theory and Application. Ph.D. Thesis, IIT Delhi, New Delhi, India, 2005.
19. Boneh, D.; Shoup, V. A Graduate Course in Applied Cryptography. Available online: https://toc.cryptobook.us/book.pdf (accessed on 1 July 2021).
20. Pierre, A.G. *Semigroups: An Introduction to the Structure Theory*; Routledge: Oxfordshire, UK, 2017.

applied sciences

Article

Assembly Sequence Planning Using Artificial Neural Networks for Mechanical Parts Based on Selected Criteria

Marcin Suszyński * and Katarzyna Peta

Institute of Mechanical Technology, Poznan University of Technology, 60-965 Poznan, Poland; katarzyna.peta@put.poznan.pl
* Correspondence: marcin.suszynski@put.poznan.pl

Abstract: The proposed model of the neural network describes the task of planning the assembly sequence on the basis of predicting the optimal assembly time of mechanical parts. In the proposed neural approach, the k-means clustering algorithm is used. In order to find the most effective network, 10,000 network models were made using various training methods, including the steepest descent method, the conjugate gradients method, and Broyden–Fletcher–Goldfarb–Shanno algorithm. Changes to network parameters also included the following activation functions: linear, logistic, tanh, exponential, and sine. The simulation results suggest that the neural predictor would be used as a predictor for the assembly sequence planning system. This paper discusses a new modeling scheme known as artificial neural networks, taking into account selected criteria for the evaluation of assembly sequences based on data that can be automatically downloaded from CAx systems.

Keywords: assembly sequence planning (ASP); modelling; artificial neural networks

Citation: Suszyński, M.; Peta, K. Assembly Sequence Planning Using Artificial Neural Networks for Mechanical Parts Based on Selected Criteria. *Appl. Sci.* **2021**, *11*, 10414. https://doi.org/10.3390/app112110414

Academic Editor: Christian W. Dawson

Received: 15 September 2021
Accepted: 3 November 2021
Published: 5 November 2021

1. Introduction

The technological assembly process is the final and the most important stage of the production process, which determines its labor consumption and the final production costs. For this reason, the development of the most favorable technology to join parts with the given conditions is a difficult task with multi-criteria but is extremely important. Optimization or improvement of assembly at the production planning stage concerns the determination of components having a direct impact on this process.

One of the most important problems at this level is the determination of the most advantageous sequence [1–5] of the assembly and components of the production cycle but also the problem of assembly line balancing (ALB) in linear systems, which in principle are also part of activities occurring at the production process stage. These issues are fundamentally related to the degree of process automation but also to the production conditions in a given enterprise. It should be emphasized that in recent times the issues of determining the assembly sequence based on artificial intelligence methods were not very frequent, despite the rapid development of this field of knowledge and the significance of the problem [2,6,7].

Planning the assembly sequence is crucial because it relates to many of its aspects, including the number of necessary tool changes, the number of assembly directions, or even the design of mounting brackets and other instrumentation, for the analyzed assembly sequence. It also has a major impact on the overall efficiency of the process. These features of the assembly process, along with many others, have a decisive impact on the efficiency of its course, but some of them may also be criteria for assessing assembly sequences for its improvement or optimization. Assembly sequence planning (ASP) consists of determining the feasibility and at the same time, finds the most advantageous, under certain criteria: order of combining assembly units, parts, and assemblies into more complex units, which leads to obtaining a final product or a product that meets all design and functional assumptions. Due to the high complexity of the issue of choosing the appropriate assembly sequence from among all acceptable choices and at the same time remaining feasible is a difficult and complex task. This is due to a large number of possible combinations of the assembly order, as the theoretical number of variants increases exponentially with the number of parts joined. In many industrial cases, when planning the assembly process, no analysis of the sequence or selection of assembly sequences is performed, and this choice is often based only on the engineering knowledge of people directly involved in planning the assembly process, although this area often contains large reserves allowing for improvement and optimizations. This state of affairs results mainly from the difficulty of evaluating even the already generated ones, due to the constraints of the constructional nature of assembly sequences.

In the literature on the subject, the assessment and selection of the most favorable sequence are made according to various criteria, depending on the specificity of plants, availability of devices, etc. Such criteria may be: assembly time, number of changes in assembly direction, number of tool changes, degree of difficulty in reaching the next process state, degree of complexity of assembly unit movements, degree of difficulty in reaching the next process state, the necessary number of reorientations of the base unit during assembly, stability of assembly units, correctness of the assembly course itself, technological production capacity, and economy of the process. Sequence evaluation criteria may also include aspects of safety, reliability, weight, operating economy, technology, ergonomics, aesthetics, or ecology. Importantly, selected data regarding the criteria for evaluating assembly sequences can be obtained automatically from CAD assembly models, for example, the direction of joining parts obtained in this way is related to the number of changes in assembly direction for a specific sequence. Very important for this process are assembly features, which also have a direct impact on the assembly order of parts. Figure 1 presents a summary of the most commonly used criteria for optimizing the assembly process in the selection of assembly sequences in the published and analyzed scientific studies.

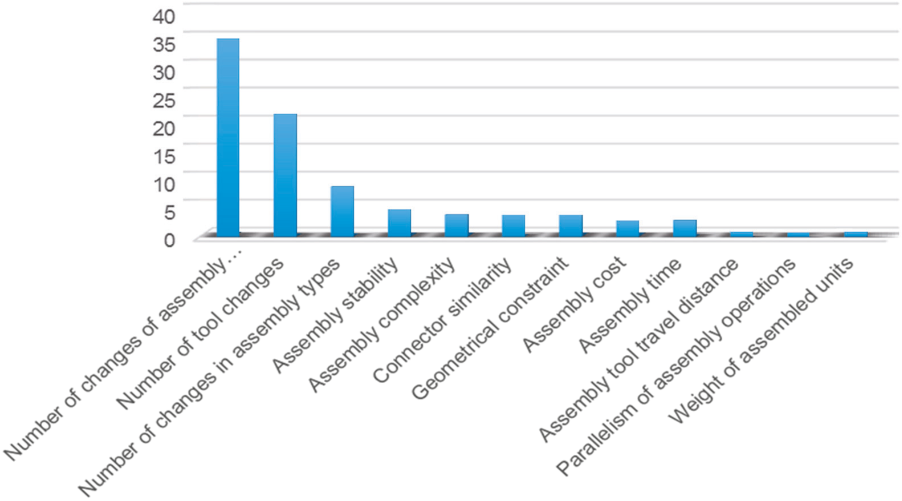


Figure 1. Criteria for the evaluation of assembly sequences in the analyzed scientific publications concerning ASP issues.

The assembly sequence planning problem belongs to a general class of optimization problems known as NP-complete. For this kind of problem, it is necessary to query the whole set of permissible solutions to ensure that the optimal assembly sequence is found. Nevertheless, because this search strategy is very time consuming and impractical in many industrial applications which are complex, have multiple criteria, and often contain issues that prove difficult to optimize, other heuristic techniques are often applied to find a solution close to the optimal one. One solution is also artificial neural networks, which is an information processing paradigm inspired by the natural biological nervous system. The very topic of assembly sequence planning using neural networks was covered in recent years by only a limited number of publications [8,9].

The input vector in neural network is multiplied by the synaptic weights, which are the weight vector. This activity is related to the implementation of the function of the postsynaptic potential and the determination of the value of the y signal calculated based on the sum of input signals multiplied by synaptic weights. The models of artificial neurons can be perceived as mathematical models. What we consider to be the first model of a neural network is the neuron model proposed by W. McCulloch and W. Pitts in 1943 and inspired by the biological model, following the pattern [9]:

$$\text{Transfer function } \Rightarrow \; y = f(\textstyle\sum_{i=1}^{k} x_i w_i + w_0) \tag{1}$$

Usually, the signal path between neurons (processing units) is as shown in Figure 2, where x_n are the neuron input signals (or the external system input data), w_n are the weights of the edge-connections (synapses), w_o is the neuron's sensitivity threshold (i.e., bias), and $f(\cdot)$ is a simple non-linear function, e.g., a sigmoid or logistic one. Activation (transfer) (AF) functions are possible for each of the hidden and output layers [3].

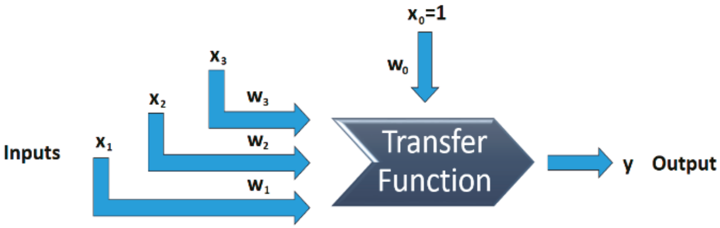


Figure 2. Schematic representation of the artificial neural network.

These studies are aimed at showing the possibility of predicting the assembly time of mechanical products based on variable factors influencing this parameter. The advantage of using artificial neural networks over other optimization algorithms is the ability to predict the assembly time without knowing the mathematical model that describes this phenomenon. This allows to obtain adequate results, also in the conditions of having incomplete production data. This procedure is indirectly aimed at indicating the assembly sequence by selecting the least time-absorbing solution. The article focuses on the application of artificial neural networks as a universal tool of artificial intelligence to support predictive tasks in the area of assembly of machine and device parts. The authors did not find any articles in which the issue of minimizing assembly time, which is important from the point of view of production efficiency, is solved with the use of art neural networks or other methods corresponding to the current trends in the use of artificial intelligence methods. Difficulties in developing an assembly time prediction procedure are mainly focused on providing an appropriate number of examples teaching the neural network. This was achieved by experimentally testing the operation of the network after each set of 100 examples was prepared. The criterion for accepting the network model for further analysis was the achievement of network efficiency during verification at a level greater than 90%. This publication should contribute to a better explanation of the relationship between the determinants of the technological process and its time consumption.

This paper discusses a modelling scheme known as artificial neural networks. The neural network approach has been used for analyzing all feasible assembly sequences. This network structure is suitable for this kind of problem. Proposed assembly planning system is a graph-based approach in the representation of product.

2. Related Works

One of the most important issues in determining the assembly sequence is the appropriate data structure, which means graph representations, mainly directed graphs or hypergraphs. This kind of structure can be considered as formalisms to encode the feasible assembly sequences. To determine all feasible sequences an appropriate graph search algorithm is necessary. The commonly used algorithm for directed graphs or hypergraphs is a heuristically guided search algorithm A*. Although exhaustive search is the simplest and most popular strategy ensuring the complete of the task, it is quite often impractical. This approach is usually used in cases where the number of parts is small (simple assembly objects). In the case when the number of parts increases, these strategies may have limitations due to the problem of combinatorial explosion.

Studies on ASP have implemented different heuristics optimization algorithms such as genetic algorithm, simulated annealing, evolutionary algorithm, ant colony optimization algorithm, and immune and other heuristic methods [10–17].

In paper [10] to solve the assembly sequence planning of a certain type of product, first of all, the rule of nomenclature is designed. Secondly, geometric feasibility and coherence are designed as constraint conditions and these two are combined with each other as the objective function. Finally, authors proposed a novel method under the name of immune particle swarm optimization algorithm. The results show that the immune particle swarm algorithm can be effective and useful in solving the problem of planning the assembly sequence.

Authors of [12] address assembly sequence planning problem and propose an improved cat swarm optimization (CSO) algorithm and redefine some basic CSO concepts and operations according to ASP characteristics. The feasibility and the stability of this improved CSO are verified through an assembly experiment and compared with particle swarm optimization.

Paper [13] proposes an ASP algorithm based on the harmony search (HS), which has an outstanding global search ability to obtain the global optimum. To solve the sequence planning problem, an improved harmony search algorithm is proposed in four aspects: (1) an encoding of harmony is designed based on ASP problems; (2) an initial harmony memory (HM) is established using the opposition-based learning (OBL) strategy; (3) a particular way to improvise a new harmony is developed; and (4) a local search strategy is introduced to accelerate the convergence speed. The proposed ASP algorithm is verified by two experiments.

In paper [17], an attempt is made to generate optimal feasible assembly sequences using design for assembly concept by considering all the assembly sequence testing criteria from obtained feasible assembly sequences. To generate all sets of assembly sequences a simulated annealing technique is used. Sequences consist of $n - 1$ levels during assembly, which are reduced by the DFA concept. DFA uses functionality of the assembled parts, material of the assembled parts, and liaison data to reduce the number of levels of the assembly by considering the directional changes as the objective function.

In this article, an assembly sequence planning system is proposed. The neutral network structure is suitable for this kind of problem. The network is capable of predicting the assembly time, which allows one to choose the best assembly sequence from all the feasible sequences.

3. Methodology

3.1. The Scope of Research Studies

The following research tasks were performed:

- Indication of determinants affecting the assembly time (number of tool changes, the number of changes in assembly direction and its stability);
- Measurement of assembly time on an example mechanical part;
- A set of prepared input and output data has been implemented in the neural network;
- Determination of constant parameters of the neural network model:

1. 3 input neurons (number of tool changes, number of changes in assembly directions, and stability of the assembly unit) and 1 output neuron (assembly time);
2. Percentage of teaching (80%), testing (10%), and verification (10%) examples;
3. Regression model (determination of the quantitative and floating-point numerical values).

- Development of the most effective model of neural network:
  1. Changing network learning algorithms (steepest gradient, scaled conjugate gradient, Broyden–Fletcher–Goldfarb–Shanno, and RBFT radial basis function teaching);
  2. Network topography (multilayer perceptron and network with radial basis functions);
  3. Activation functions (linear, sigmoidal, exponential, hyperbolic, and sine);
  4. Number of hidden neurons (1–12).
- Selection of the most effective network model, taking into account the error of the sum of squared differences generated by the network;
- Introduction of previously untested data to the network, allowing verification of the effectiveness of prediction of assembly time.

This methodology first defined general research tasks and then performed network testing based on a specific example of a mechanical part. The graphic concept of the proposed method is presented in Figure 3.

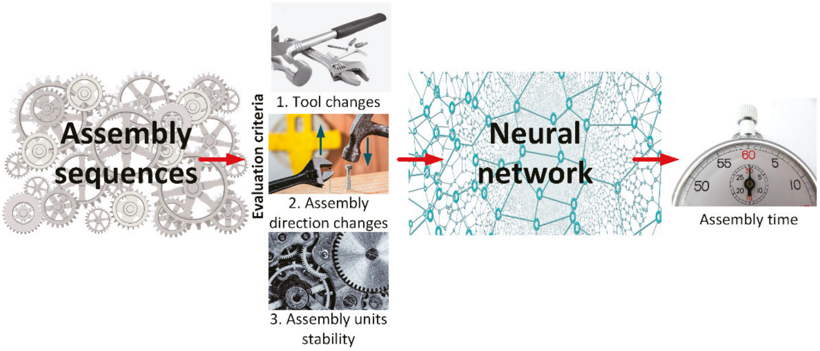


Figure 3. The graphic concept of the proposed method of prediction of assembly time.

In the future, this research may be extended to the verification of other parts of machines and devices based on the developed model of the neural network. The selected criteria determining the assembly time are universal and it is assumed that they are also adequate to other solutions.

3.2. Assessment Criteria for the Assembly Sequence

The proposed tool, based on artificial neural networks, has the objective to support the determined sequence for manual assembly (although it is also possible to apply it, albeit after modifications, to an automated process). It was assumed that at the current stage of research it is used in a specific mechanical production company, where the conditions of the assembly process for newly introduced products are subject to ASP analysis, and the processes implemented were used to teach the network. This applies to issues related to, for example, the available machine park, production organization, process control and supervision, or the level of training of employees, especially in the aspect of manual assembly.

The following assembly sequence evaluation criteria were used as input to the process:

- Number of tool changes for the respective assembly sequence.
  This criterion indicates the number of tool changes during assembly operations. Operation constitutes the main structural element of a technological assembly process. In

this work, operations should be understood as, for example, activities such as riveting, drilling, fitting, and screwing, which are related to changing tools. Depending on the type of parts to be installed, the required tools can be assigned to them in a simple manner, from the set of tools utilized in the considered assembly process.

- The number of changes in assembly direction for the respective assembly sequence.
  It is the most frequent optimization criterion in ASP. This criterion is connected with the direction in which the parts are attached during their assembly. There are 6 main assembly directions, along the 3 main axes: ± X, ± Y, and ± Z.
- Number of stable and unstable units for the specific assembly unit.
  Stability criterion determines the number of stable and unstable units for a particular assembly sequence. We assume that a stable unit is such a unit that remains in an assembled state, regardless of the force applied to it. The applied forces may be the force of gravity or the forces associated with the movement of parts or an assembly unit.

We justify the adoption of these criteria for the evaluation, among others, with the fact that, as one of the few, they can be automatically obtained from the CAD assembly model, although it is also assumed that the data can be completed manually.

The purpose of the system is to assist in the estimation of time for all acceptable sequences under constructional constraints (i.e., feasible ones) and thus enable the selection of the most favorable one under existing manufacturing conditions. Under these evaluation criteria it is the sequence with the lowest number of tool changes, the smallest number of changes in assembly directions, and the smallest possible number of unstable states that will likely be indicated as the most favorable one; however, it is practically impossible to obtain such values with these criteria for a single sequence. This is related, for example, to the weights of individual criteria in relation to the specific assembly process.

3.3. Neural Network Assumptions

Artificial neural networks were used to evaluate the sequence of combining assembly units. For this purpose, the input and output features of the network were selected and a set of teaching examples was prepared. The input data were the number of tool changes, the number of changes in the assembly direction, and assembly stability, while the assembly time was classified as the group of output data. An important task is to provide an appropriate number of training samples and identify connections between data, which when combined allow for obtaining sufficient results and network efficiency [18]. In order to prepare the training dataset, the numerical values of individual features were normalized, allowing one to obtain independence between all analyzed data and to ensure equivalence. The numerical values of the features initially appearing in different ranges were scaled to values in the range <0.1> using a linear transformation. The task of data normalization was performed by the min-max function, calculating the difference between the scaled value and minimum value and scaling it by the range of numerical data according to the formula:

$$X^* = \frac{X - min(X)}{\max(x) - min(x)} \tag{2}$$

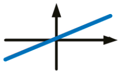
To obtain adequate efficiency, neural network training is performed, consisting of minimizing the prediction error function determined by the sum of squares (*SOS*) as defined by the formula:

$$SOS = \sum_{i=1}^{n} (y_i - y_i^*)^2 \tag{3}$$

where: n is the number of training examples, y_i is an expected network output value, and y_i^* is an actual network output value.

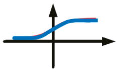
The error surface is paraboloid-shaped with one distinct minimum, it is associated with the neurons belonging to the output layer, and it is calculated after each epoch—repeating the training algorithm. The error is related to a discrepancy between the values obtained at the network output and the reference values included in the training dataset. Errors are also determined for neurons in hidden layers by backpropagation, which consists of adjusting the weight values depending on the assessment of the neuron error in a multilayer network, using gradient optimization methods. The error backpropagation algorithm is implemented in the direction from the output layer to the input layer, which is the opposite direction to the information flow. The effectiveness of a neural network is directly related to the error function and is calculated as the ratio of correctly classified or approximated cases to all cases included in the dataset. In order to obtain the highest efficiency of prediction, the parameters describing the neural network model were changed and empirically selected: the number of layers (input, output, and hidden) and the included neurons, the presence of an additional neuron—bias and network learning rules, including the learning algorithm and activation function. The input layer consists of neurons to which the input signals are sent to the first hidden layer. The set of input data is divided into three groups: (1) training data string that allows reflection on prediction tasks, (2) test data, which check the operation of the network, (3) verification data, which evaluate the network performance based on new, previously unused set of numerical data. The number of neurons and hidden layers is selected empirically, enabling a compromise between its extensive structure and the correct generalization of the processed data. The output layer of the network is a collection of neurons representing the output signals. The number of neurons in the output layer is identical to the number of output data points constituting the result of the network. In addition, in the model of the neural network there may be an additional neuron bias, called the artificial signal generator, constituting an additional input for the neuron with a value of +1 and improving the stability of the network during the training process. The effectiveness of the network is determined by the activation function of hidden and output neurons, which take the following form: linear (directly transmitting the excitation value of the neuron to the output), logistic (sigmoidal curve with values greater than 0 and less than 1), exponential (with a negative exponent), and hyperbolic (hyperbolic tangent curve with values greater than −1 and less than 1). To verify the threshold value of the input signal needed to activate the neuron, the activation functions *f(x)* are used:

- Linear with output values in the range from $-\infty$ to ∞:



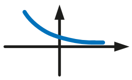
$$f(x) = x \tag{4}$$

- Logistic (sigmoidal) with output values in the range from 0 to 1:



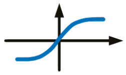
$$f(x) = \frac{1}{1-e^{-1}} \tag{5}$$

- Exponential with output values in the range from 0 to ∞:



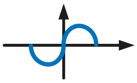
$$f(x1) = e^{-x} \tag{6}$$

- Hyperbolic (hyperbolic tangent) with output values in the range from -1 to 1:



$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{7}$$

- Sine with output values from the range from −1 to 1:



$$f(x) = \sin(x) \tag{8}$$

The selection of the neural network learning algorithm affects its effectiveness. The general principle of the learning algorithms is to minimize the error function by iteratively modifying the weights assigned to neurons. The learning process involves entering successive learning cases containing information and correct network responses to a set of input values. The iterative algorithm is stopped when the ability to generalize the learning results deteriorates. There are many neural network learning algorithms. In this study, the methods of steepest descent, gradient scaling, and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm were used. In the steepest descent method, after specifying the search direction, the minimum value of the function in this direction is determined, as opposed to the simple gradient method, which uses a shift with a constant step. An important feature of the steepest descent method is that each new direction towards the function optimum is orthogonal to the previous one. Movement in one direction continues until this direction turns out to be tangent to a certain line of constant value of the objective function. The principle of the steepest slope, when designating subsequent search directions, requires carrying out a large number of searches along the successively proposed straight lines. In this situation, a neural network teaching method based on conjugate directions is a better solution. The algorithm determines the appropriate direction of movement along the multidimensional error surface. Then a straight line is drawn over the error surface in this direction and the minimum value of the error function is determined for all points along the straight line. After finding the minimum value along the initially given direction, a new search direction is established from this minimum and the whole process is repeated. Accordingly, there is a constant shift towards decreasing values of the error function until a point is found which corresponds to the function minimum. The second derivative determined in this direction is set to zero during the next learning steps. To maintain the second derivative value of zero, the direction's conjugate to the previously chosen direction is determined. Moving in the conjugate direction does not change the fixed (zero) value of the second derivative computed along the previously selected direction. Determining the conjugate direction is associated with the assumption that the error surface has a parabolic shape. The Broyden–Fletcher–Goldfarb–Shanno algorithm refers to a quasi-Newton algorithm that modifies the weights of the interneural connections after each epoch based on the mean error gradient. The principle of operation is based on the search for the minimum squared error function with the use of a Hessian matrix (a matrix of partial derivatives of the second-order), the inverse of which is generated by an algorithm that initially uses the steepest descent method, and in the next step it refers to the estimated Hessian. For radial networks, standard learning procedures are used, including k-means center determination, k-neighbor deviation, and then output layer optimization. The k-means method is a method that consists of finding and extracting groups of similar objects (clusters). Thus, k different clusters are created; the algorithm allows one to move objects from one cluster to another until the variations within and between clusters are optimized. The similarity of data in a cluster is supposed to be as large as possible and separate clusters should differ as much as possible from each other. In the k-neighbor method, each dataset is assigned a set of n values that characterize it and then placed in an n-dimensional space. Assigning data to an existing group consists of finding the k-nearest objects in n-dimensional space and then selecting the most numerous group.

The different types of neural network topologies differ in structure and operating principles, the basis of which are the multilayer perceptron (MLP) and the network with radial basis functions (RBF). The multilayer perceptron consists of many neurons arranged in layers that calculate the sum of the inputs, and the determined excitation level is an argument of the activation function and then the calculated network output value. All neurons are arranged in a unidirectional structure in which the transmission of signals takes place in a strictly defined direction—from input to output. A key task in MLP network design is to determine the appropriate number of layers and neurons, usually performed empirically. A network with radial base functions often has only one hidden layer, containing radial neurons having a Gaussian character. On the other hand, a simple linear transformation is usually applied to the output layer. The task of radial neurons is to recognize the repetitive and characteristic features of input data groups.

In order to elaborate on the best model of the network, a number of constant and variable parameters were determined, tested by the multiple random sampling method, resulting in 10.000 network variants. The error of the sum of squared differences generated for each set of test parameters was established as the criterion for network effectiveness. The constant parameters of the artificial neural network are:

1. 3 input neurons (number of tool changes, number of changes in assembly directions, and stability of the assembly unit) and 1 output neuron (assembly time);
2. Percentage of teaching (80%), testing (10%), and verification (10%) examples;
3. Regression model (determination of the quantitative and floating-point numerical values).

Variable network parameters that were altered randomly during the generation of network models were:

1. Network learning algorithms (steepest gradient, scaled conjugate gradient, Broyden–Fletcher–Goldfarb–Shanno, and RBFT radial basis function teaching);
2. Network topography (multilayer perceptron and network with radial basis functions);
3. Activation functions (linear, sigmoidal, exponential, hyperbolic, and sine);
4. Number of hidden neurons (1–12).

4. Results and Discussion

4.1. Product Structure and Results

The structure of the product intended for assembly is presented in the form of a modified directed graph of assembly states. Moreover, we assumed that parts in the directed graph (digraph) (or the assembled units) are marked as vertices, while the directed edges demonstrate the possible sequences (paths) for assembling them. It is further assumed that the assembly of further elements takes place by adding a part or subassembly consisting of more parts (treated as a single assembled part) to the nth stage assembly. The directed edges connecting the vertices contain information about the stability of the newly formed assembly state, the direction of attachment of the parts, and the tool applied. The described digraph can be generated automatically, based on the CAD assembly drawing.

The basis for executing ASP according to the defined criteria for a specific assembly process is the determination of all assembly sequences that are feasible due to constraints of a constructional nature. The matrix record (e.g., in the form of an assembly states matrix or an assembly graph matrix) of assembly units enables us to determine all variants of assembly sequences using the appropriate algorithm (this procedure is not discussed here and it is reduced to finding all the paths in the digraph leading from the starting vertex x_s, constituting the base part, to the final vertex x_e, i.e., the last state of the assembled product—$x_s, \ldots, x_e$).

The task of determining the sequence of assembly using artificial neural networks was performed for a sample product—a forklift door, consisting of eight main assembly units:

- 1: door welded construction;
- 2: lock;
- 3: cassette lock;
- 4: door reinforcement bar with passenger's handle;
- 5: lock cover;
- 6: door seal;
- 7: lower glass;
- 8: upper glass.

In the first stage, using the construction documentation, the base part was determined in the form of assembly unit no. 1. Then, a digraph of the structural limitations of the assembly states was constructed, shown in Figure 4.

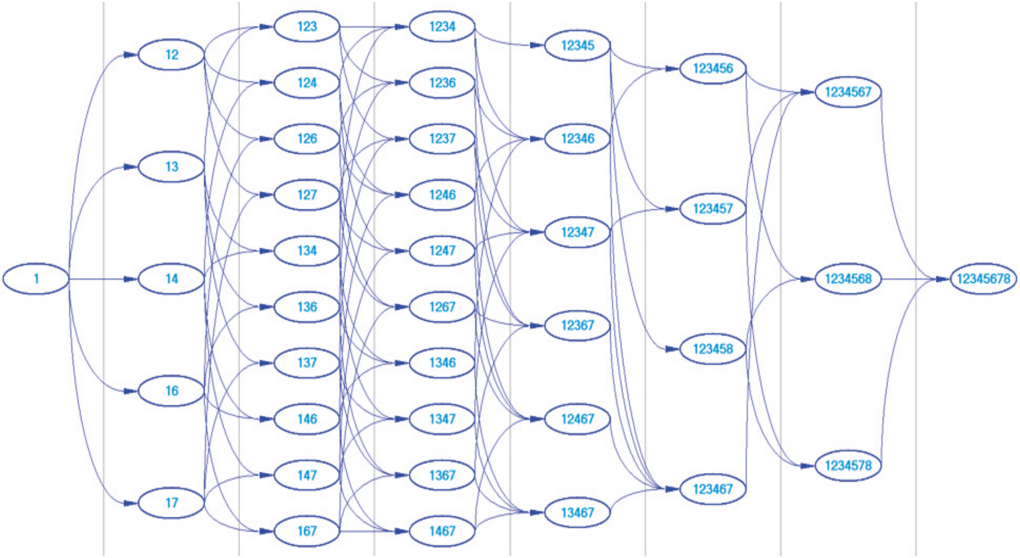


Figure 4. Digraph of the structural constraints of the forklift door assembly states.

It was assumed that the assembly of subsequent units takes place by adding another assembly unit to the assembly state of the nth stage. Based on the constructed digraph recorded in the form of the assembly state matrix, we determined, with the use of a selected graph search algorithm 252, those assembly sequences that were possible under the constraints of the structural nature (Table 1) [19,20], which constitute the basis for further analysis.

Table 1. Selected feasible assembly sequences generated due to design constraints.

No.	Start	1	2	3	4	5	6	STOP	RESULTING SEQUENCE
1	1	12	123	1234	12345	123456	1234567	12345678	12345678
2	1	12	123	1234	12345	123456	1234568	12345678	12345687
3	1	12	123	1234	12345	123457	1234567	12345678	12345768
4	1	12	123	1234	12345	123457	1234578	12345678	12345786
5	1	12	123	1234	12345	123458	1234568	12345678	12345867
6	1	12	123	1234	12345	123458	1234578	12345678	12345876
7	1	12	123	1234	12346	123456	1234567	12345678	12346578
8	1	12	123	1234	12346	123456	1234568	12345678	12346587
9	1	12	123	1234	12346	123467	1234567	12345678	12346758
10	1	12	123	1234	12347	123457	1234567	12345678	12347568
11	1	12	123	1234	12347	123457	1234578	12345678	12347586
12	1	12	123	1234	12347	123467	1234567	12345678	12347658
13	1	12	123	1236	12346	123456	1234567	12345678	12364578
14	1	12	123	1236	12346	123456	1234568	12345678	12364587
15	1	12	123	1236	12346	123467	1234567	12345678	12364758
...	...	...	...	...	...	...	...	...	...
252	1	17	167	1467	13467	123467	1234567	12345678	17643258

Table 2 presents the most effective neural networks for predicting the assembly time of the discussed product. By assessing the values of the sum of squared differences error and the effectiveness of the selected neural networks, it was found that the best results were obtained for network no. 9—the 3-8-1 RBF (Figure 5). We selected it for further analysis (a network with radial basis functions with three input, eight hidden, and one output neurons), in which hidden neurons were activated by a Gaussian function, and output neurons by a linear one, obtaining about 99% efficiency for the group of verification data.

Table 2. Values of neural network parameters that were found best for prediction of assembly time.

Network No.	Network Name	Effectiveness (Learning)	Effectiveness (Testing)	Effectiveness (Verification)	SOS Error (Learning)	SOS Error (Testing)	SOS Error (Verification)	The Learning Algorithm	Activation (Hidden Neurons)	Activation (Output Neurons)
1	RBF 3-7-1	0.4146	0.7848	0.9926	0.0229	0.0587	0.0050	RBFT	Gaussian	Linear
2	RBF 3-9-1	0.4381	0.7643	0.9958	0.0241	0.0698	0.0112	RBFT	Gaussian	Linear
3	RBF 3-8-1	0.4050	0.9764	0.9929	0.0231	0.0456	0.0033	RBFT	Gaussian	Linear
4	RBF 3-2-1	0.0794	0.9668	0.9913	0.0274	0.0636	0.0090	RBFT	Gaussian	Linear
5	RBF 3-7-1	0.4516	0.9759	0.9925	0.0220	0.0574	0.0042	RBFT	Gaussian	Linear
6	RBF 3-2-1	0.0794	0.9668	0.9913	0.0274	0.0636	0.0090	RBFT	Gaussian	Linear
7	RBF 3-2-1	0.0794	0.9668	0.9913	0.0274	0.0636	0.0090	RBFT	Gaussian	Linear
8	RBF 3-2-1	0.0794	0.9668	0.9913	0.0274	0.0636	0.0090	RBFT	Gaussian	Linear
9	RBF 3-8-1	0.4522	0.9778	0.9942	0.0220	0.0574	0.0042	RBFT	Gaussian	Linear
10	RBF 3-6-1	0.4207	0.8487	0.9981	0.0227	0.0567	0.0038	RBFT	Gaussian	Linear

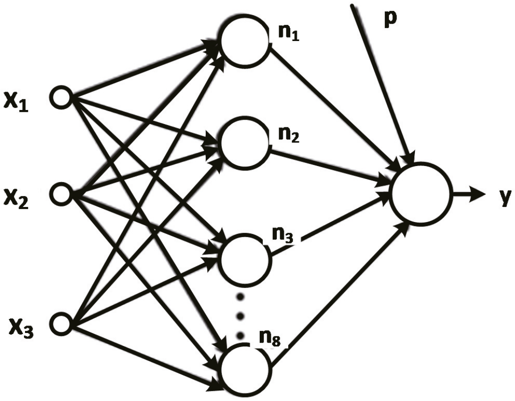


Figure 5. RBF network model (x_1 is the number of tool changes, x_2 is the number of changes in assembly directions, x_3 is a stability of the assembly unit, n1–n8 are hidden neurons, p is a bias, and y is an assembly time).

Figure 6 presents the changes in the value of the learning error of the selected RBF network depending on the number of learning cycles. The neural network was found in the first learning cycle—after the first iteration of the training algorithm. The stabilization of the error value occurred in the sixth learning cycle.

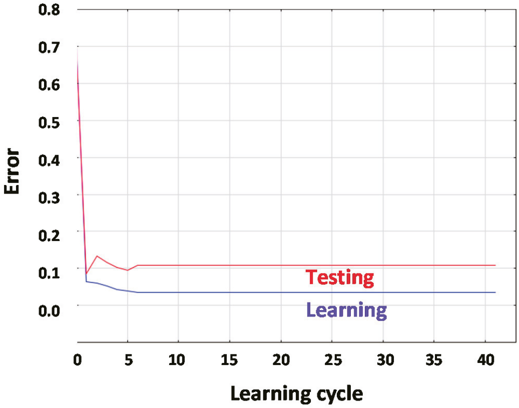


Figure 6. Changes in the value of network learning errors depending on the number of learning cycles.

In the learning process of the neural network, the weight values for all neurons are adjusted. This has an impact on the obtained results because the weights can weaken (negative values) or strengthen (positive values) the signals transferred by individual layers of the network. Table 3 presents the weight values generated for the analyzed RBF network.

Table 3. Neural network weights for prediction of assembly time and network parameters that were found best for prediction of assembly time.

Connections RBF 3-8-1	Weight Values	Connections RBF 3-8-1	Weight Values	Connections RBF 3-8-1	Weight Values
X_1—hidden neuron 1	0.400000	X_3—hidden neuron 5	1.000000	Radial range hidden neuron 5	0.640312
X_2—hidden neuron 1	0.500000	X_1—hidden neuron 6	0.600000	Radial range hidden neuron 6	0.200000
X_3—hidden neuron 1	1.000000	X_2—hidden neuron 6	0.500000	Radial range hidden neuron 7	0.200000
X_1—hidden neuron 2	0.00000	X_3—hidden neuron 6	1.000000	Radial range hidden neuron 8	0.200000
X_2—hidden neuron 2	0.00000	X_1—hidden neuron 7	0.400000	Hidden neuron 1—y	0.044928
X_3—hidden neuron 2	1.000000	X_2—hidden neuron 7	0.00000	Hidden neuron 2—y	−0.059589
X_1—hidden neuron 3	0.200000	X_3—hidden neuron 7	1.000000	Hidden neuron 3—y	−0.006650
X_2—hidden neuron 3	0.00000	X_1—hidden neuron 8	0.400000	Hidden neuron 4—y	0.074476
X_3—hidden neuron 3	1.000000	X_2—hidden neuron 8	0.500000	Hidden neuron 5—y	0.254939
X_1—hidden neuron 4	0.600000	X_3—hidden neuron 8	1.000000	Hidden neuron 6—y	−0.094136
X_2—hidden neuron 4	0.500000	Radial range hidden neuron 1	0.200000	Hidden neuron 7—y	0.006649
X_3—hidden neuron 4	1.000000	Radial range hidden neuron 2	0.200000	Hidden neuron 8—y	−0.045479
X_1—hidden neuron 5	1.000000	Radial range hidden neuron 3	0.200000	Data offset—y	0.541008
X_2—hidden neuron 5	1.000000	Radial range hidden neuron 4	0.200000		

Table 4 summarizes the actual and expected assembly time prediction values, whereas Figure 7 is a graphical interpretation of their dependencies. A set of verification data containing previously unused input and output data was selected for the analysis. The results of the analysis confirm the effectiveness of the prediction performed by the RBF 3-8-1 neural network. The obtained results, both expected and obtained at the network output, are comparable. The operation of the network was tested on the basis of 10 random assembly sequences and the result of assembly time was obtained for each of them. Based on the results presented in Table 4, it can be indicated which of the assembly sequences was characterized by the shortest assembly time, which is therefore the optimal solution.

Table 4. Assembly time values expected and obtained at the network output.

Case No.	Expected Network Value	Network Output Value
1	0.645	0.628
2	0.635	0.598
3	0.573	0.531
4	0.595	0.610
5	0.525	0.508
6	0.656	0.689
7	0.629	0.661
8	0.595	0.619
9	0.620	0.597
10	0.532	0.559

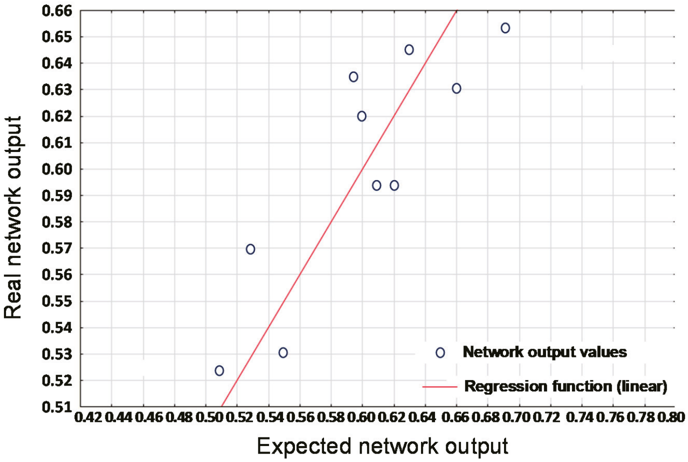


Figure 7. Comparison of the expected and obtained assembly time values at the network output.

4.2. Discussion

Presented method proposes the selection of the best assembly sequence based on the estimated assembly time for the selected product. It works on the basis of selected universal criteria for the evaluation of assembly sequences and their impact on the process time. In principle, its correct operation is based on constant production conditions, which is a prerequisite for its proper operation and correctness of the network learning process. Universal criteria for assessing the assembly sequence proposed in this paper can be effectively automatically retrieved from CAD documentation, although this is not the subject of the presented analysis. The obtained test results confirm that it is possible to develop procedures supporting the determination of the assembly sequence of mechanical products. The neural network model effectively predicts the time of the assembly process. Further research should focus on developing a more universal method and increasing the amount of data to enable network learning.

The effectiveness of the method depends mainly on the number of cases teaching the neural network that are able to generalize the knowledge and the neural network to different products to be assembled. At the moment, the effectiveness of the network in the data verification group is 99%. Entering new data into the network will improve the efficiency of the time sensitive tasks and universally the possibility of applying the procedure to new, not considered cases.

Thus, a network constraint may be a greater number of errors when predicting assembly time for other products. The aim of the authors is to develop the conducted research and verify the operation of the network on a wide range of products. A neural network model was developed to meet the requirements of all mechanical parts. The goal was to develop an overall model. Then, its effectiveness was verified on the basis of one selected product—the door of a forklift truck.

5. Conclusions

The article describes a mechanical assembly time prediction system operating in a neural network, determined by the criteria: the number of tool changes, the number of assembly direction changes, or the stability of the assembly units. The principle of operation and training of the network is its work in a specific mechanical production period; it allows one to determine the most advantageous workplace configuration, production organization, process control, or level of employee training. It is necessary for the best possible network search results.

The obtained results of the analyses confirmed the effectiveness of the previously developed model. The authors assumed that it would also be suitable for other mechanical products, and further studies will be carried out to prove these assumptions. The development of a universal model for selecting the least time-consuming assembly sequence will make it possible to improve many assembly processes. This is of particular importance for products consisting of many parts and in complex manufacturing processes.

The obtained test results confirm that it is possible to develop procedures supporting the determination of the assembly sequence of mechanical products. The model of the neural network, containing universal criteria determining the time of the assembly process, was verified on the example of the assembly of the door of a forklift truck, confirming its effectiveness. Further research should focus on checking the usefulness of the neural network also for other mechanical products. The effectiveness of the method depends mainly on the number of cases teaching the neural network that are able to generalize the knowledge and the neural network to different products to be assembled. Thus, a network constraint may be a greater number of errors when predicting assembly time for other products. The aim of the authors is to develop the conducted research and verify the operation of the network on a wide range of products.

Author Contributions: Conceptualization, M.S.; methodology, K.P. and M.S.; software, K.P. and M.S.; validation, M.S. and K.P.; formal analysis, M.S. and K.P.; investigation, M.S. and K.P.; data curation, M.S.; writing—original draft preparation, M.S. and K.P.; writing—review and editing, K.P. and M.S.; visualization, K.P. and M.S.; supervision, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Ministry of Science and Higher Education of Poland (No. 0614/SBAD/1529).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, G.A.; Bahubalendruni, M.R.; Prasad, V.V.; Ashok, D.; Sankaranarayanasamy, K. A novel Geometric feasibility method to perform assembly sequence planning through oblique orientations. *Eng. Sci. Technol. Int. J.* **2021**. [CrossRef]
2. Su, Y.; Mao, H.; Tang, X. Algorithms for solving assembly sequence planning problems. *Neural Comput. Appl.* **2021**, *33*, 525–534. [CrossRef]
3. Butlewski, M.; Suszyński, M.; Czernecka, W.; Pajzert, A.; Radziejewska, M. Ergonomic criteria in the optimization of as-sembly processes, Cristina Feniser. In *Proceedings of the 6th RMEE2018—Performance Management or Management Performance—Todesco*; Publishing House: Cluj-Napoca, Romania, 2018; pp. 424–431.
4. Sąsiadek, M. Planning and analysis of mechanical assembly sequences in design engineering—Part I: The Method. *Teh. Vjesn. Tech. Gaz.* **2015**, *22*, 337–342. [CrossRef]

5. Wu, W.; Huang, Z.; Zeng, J.; Fan, K. A decision-making method for assembly sequence planning with dynamic resources. *Int. J. Prod. Res.* **2021**, 1–20. [CrossRef]
6. Watanabe, K.; Inadaa, S. Search algorithm of the assembly sequence of products by using past learning results. *Int. J. Prod. Econ.* **2020**, *226*, 107615. [CrossRef]
7. Zhang, H.; Peng, Q.; Zhang, J.; Gu, P. Planning for automatic product assembly using reinforcement learning. *Comput. Ind.* **2021**, *130*, 103471. [CrossRef]
8. Chen, W.C.; Tai, P.H.; Deng, W.J.; Hsieh, L.F. A three-stage integrated approach for assembly sequence planning using neural networks. *Expert Syst. Appl.* **2008**, *34*, 1777–1786. [CrossRef]
9. Sinanoğlu, C.; Börklü, H.R. An assembly sequence-planning system for mechanical parts using neural network. *Assem. Autom.* **2005**, *25*, 38–52. [CrossRef]
10. Zhang, H.; Liu, H.; Li, L. Research on a kind of assembly sequence planning based on immune algorithm and particle swarm optimization algorithm. *Int. J. Adv. Manuf. Technol.* **2013**, *71*, 795–808. [CrossRef]
11. Biswal, B.B.; Pattanayak, S.K.; Mohapatra, R.N.; Parida, P.K.; Jha, P. Generation of optimized robotic assem-bly sequence using immune. In Proceedings of the ASME, International Mechanical Engineering Congress and Exposition, Houston, TX, USA, 9–15 November 2012; pp. 1–9.
12. Guo, J.; Sun, Z.; Tang, H.; Yin, L.; Zhang, Z. Improved Cat Swarm Optimization Algorithm for Assembly Sequence Planning. *Open Autom. Control. Syst. J.* **2015**, *7*, 792–799. [CrossRef]
13. Li, X.; Qin, K.; Zeng, B.; Gao, L.; Su, J. Assembly sequence planning based on an improved harmony search algorithm. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 2367–2380. [CrossRef]
14. Wang, D.; Shao, X.; Liu, S. Assembly sequence planning for reflector panels based on genetic algorithm and ant colony optimization. *Int. J. Adv. Manuf. Technol.* **2017**, *91*, 987–997. [CrossRef]
15. Xin, L.; Shang, J.; Cao, Y. An efficient method of automatic assembly sequence planningfor aerospace industry based on genetic algorithm. *Int. J. Adv. Manuf. Technol.* **2017**, *90*, 1307–1315. [CrossRef]
16. Zeng, C.; Gu, T.; Zhong, Y.; Cai, G. A Multi-Agent Evolutionary algorIthm for Connector-Based Assembly Sequence Planning. *Procedia Eng.* **2011**, *15*, 3689–3693. [CrossRef]
17. Murali, G.B.; Deepak, B.B.V.L.; Bahubalendruni, M.V.A.R.; Biswal, B.B. Optimal Assembly Sequence Planning To-wards Design for Assembly Using Simulated Annealing Technique. *Res. Des. Communities* **2017**, *1*, 397–407.
18. Tadeusiewicz, R.; Chaki, R.; Chaki, N. *Exploring Neural Networks with C#*; CRC Press: Boca Raton, FL, USA, 2015.
19. Suszyński, M.; Żurek, J. Computer aided assembly sequence generation. *Manag. Prod. Eng. Rev.* **2015**, *6*, 83–87. [CrossRef]
20. Suszyński, M.; Żurek, J.; Legutko, S. Modelling of assembly sequences using hypergraph and directed graph. *Teh. Vjesn. Tech. Gaz.* **2014**, *21*, 1229–1233.

symmetry

MDPI

Article

Mechanical Assembly Sequence Determination Using Artificial Neural Networks Based on Selected DFA Rating Factors

Marcin Suszyński [1,]*, Katarzyna Peta [1], Vít Černohlávek [2] and Martin Svoboda [2]

[1] Institute of Mechanical Technology, Poznan University of Technology, 60-965 Poznan, Poland; katarzyna.peta@put.poznan.pl
[2] Faculty of Mechanical Engineering, University of Jan Evangelista Purkyně in Ústí nad Labem, 400 96 Ústí nad Labem, Czech Republic; vit.cernohlavek@ujep.cz (V.Č.); martin.svoboda@ujep.cz (M.S.)
* Correspondence: marcin.suszynski@put.poznan.pl

Abstract: In this paper, an assembly sequence planning system, based on artificial neural networks, is developed. The problem of artificial neural network itself is largely related to symmetry at every stage of its creation. A new modeling scheme, known as artificial neural networks, takes into account selected DFA (Design for Assembly) rating factors, which allow the evaluation of assembly sequences, what are the input data to the network learning and then estimate the assembly time. The input to the assembly neural network procedure is the sequences for assembling the parts, extended by the assembly's connection graph that represents the parts and relations between these parts. The operation of a neural network is to predict the assembly time based on the training dataset and indicate it as an output value. The network inputs are data based on selected DFA factors influencing the assembly time. The proposed neural network model outperforms the available assembly sequence planning model in predicting the optimum assembly time for the mechanical parts. In the neural networks, the BFGS (the Broyden–Fletcher–Goldfarb–Shanno algorithm), steepest descent and gradient scaling algorithms are used. The network efficiency was checked from a set of 20,000 test networks with randomly selected parameters: activation functions (linear, logistic, tanh, exponential and sine), the number of hidden neurons, percentage set of training and test dataset. The novelty of the article is therefore the use of parts of the DFA methodology and the neural network to estimate assembly time, under specific production conditions. This approach allows, according to the authors, to estimate which mechanical assembly sequence is the most advantageous, because the simulation results suggest that the neural predictor can be used as a predictor for an assembly sequence planning system.

Keywords: design for assembly; artificial neural networks; assembly

Citation: Suszyński, M.; Peta, K.; Černohlávek, V.; Svoboda, M. Mechanical Assembly Sequence Determination Using Artificial Neural Networks Based on Selected DFA Rating Factors. *Symmetry* **2022**, *14*, 1013. https://doi.org/10.3390/sym14051013

Academic Editors: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Received: 11 April 2022
Accepted: 12 May 2022
Published: 16 May 2022

1. Introduction

Symmetry is an issue that largely affects artificial neural networks. Symmetry can be found in the structure of the neural network itself; symmetric weight and many other related issues are also in the presented article.

The process of assembling parts into assembly units until the final product is achieved is one of the most important stages of the production process. During the process, specific features characterizing the product are created. This process also makes a major contribution to the product development itself. Planning the assembly process requires a series of analyses, i.e., the separation of assembly units and the determination of the relationships between them. The result of these analyses is the selection of the base parts for the separate assembly units and the specific assembly sequences, which are the basis for estimating assembly difficulties. Product quality and manufacturing costs depend mainly on the product structure; this structure describes the functionally imposed layout and the geometrically possible assembly sequences. It is natural to choose which one allows us to obtain a finished product in the shortest time.

Obtaining the most advantageous technology of assembling parts in given production conditions is an extremely important, multi-criteria and difficult to model task, which is mainly based on determining the assembly sequence plan (ASP) and selecting all the components of the production process, but also applies to balancing the assembly line (ALB). These issues are fundamentally related to the degree of automation of the considered process, but also to the production conditions in a given enterprise. Due to the complexity of the assembly sequence planning problem, its optimization is required in order to obtain an efficient computational approach. The aim is therefore to include the evaluation process and selecting the assembly sequence as early as possible in the product development phase. The high complexity of this process (great number of components) makes it difficult to determine the set of permissible assembly sequences and leads to a phenomenon that is difficult to solve, "combinatorial explosion"; therefore, one of the solutions to this problem is the heuristic approach presented in this publication, where neural networks based on selected DFA rating factors describing assembly sequences allow us to forecast a mechanical assembly time.

The paper, one by one, presents the optimization approach to the problem of assembly sequence generation with the use of heuristic methods, the issues of assembly sequence evaluation criteria and artificial neural networks in an assembly sequence planning problem. The most important part of the work is the concept of the system supporting the determination of the assembly sequence based on artificial neural networks and DFA rating factors for manual mechanical assembly.

2. Optimization Approach

Bio-inspired algorithms are quite modern, increasingly used, and efficient tools for the industry. However, from a mathematical point of view, these problems belong to the most general of non-linear optimization problems, where these tools cannot guarantee that the best solution will be found. The numerical cost and the accuracy of these algorithms depend, among others, "on the initialization of their internal parameters which may themselves be the subject of parameter tuning according to the application" [1].

In the literature on the subject, various optimization algorithms are proposed to optimize the ASP problem. The most common classes of algorithms used to optimize ASP are meta-heuristic approaches. The meta-heuristic algorithm provides a solution framework to optimize different optimization problems with relatively few modifications to adapt them to a specific problem and limits the general computation time. This approach turns out to be, in many cases, sufficient due to the quality of the obtained results at appropriate costs in such a multi-criteria task as assembly sequence planning, where the set of feasible solutions to be analyzed is large [2–4].

Studies on ASP have implemented different heuristic optimization algorithms, such as genetic algorithm (GA) [5–7]. The authors of [6] presented the application of genetic algorithms (GA) and the ant colony algorithm (ACO), using the example of reflector antenna assembly. The accurate simulation of the assembly was required to evaluate and optimize the ASP. The initial population was created by ACO and optimized by GA operators to generate an optimal solution. By releasing the information on the optimal solution to the ant search paths of ACO, convergence towards a globally optimal solution was accelerated. A model of the finite element simulation was used to simulate the dynamic assembly process of the reflectors according to the algorithm results of the proposed approach, which can improve the quality of the optimal solution and reduce the probability of finding a local optimal solution.

Another method based on a genetic algorithm [7] was used in the process of planning the assembly sequence for satellites in the space industry. This method takes into account a process with a large number of connections in a multi-stage and parallel assembly. Priority relationships are established between the assembly units, and the assembly sequence is represented by a directed acyclic graph. Original, a two-part crossover and mutation operators for assembling sequence were proposed.

Another approach to generate acceptable assembly sequences is an algorithm based on a simulated annealing (SA) process [8,9]. The method is derived from an analogy with thermodynamics, and more specifically, with how a liquid solidifies to form a crystal structure. In [8], the authors proposed a novel method under the name of genetic simulated annealing algorithm (GSAA) and ant colony optimization (ACO) algorithm for assembly sequence planning (ASP), which assist the planner in generating an effective assembly sequence with respect to a large constraint assembly perplexity.

The ant colony optimization (ACO) algorithm was also presented in [6,8,10–12]. In [12], an improved ASP-oriented ant colony algorithm was proposed in order to achieve an optimal or close to optimal assembly sequence. In this algorithm, the assembly operation constraint is introduced into the state transfer function as heuristic information, while the feasible transition area is set up by obtaining assembly relationships of the components. In addition, a dynamic change in pheromone trail persistence was also used.

Evolutionary algorithms (EA) for connector-based assembly sequence planning were also analyzed [13], but they generated many erroneous searches and it was necessary to build a multi-agent connector-based evolution algorithm. Competition, crossover-mutation and learning were designed as the behaviors of agents that locate a lattice-like structure environment. The presented metaheuristic algorithms are highly efficient and seem to be interesting tools in solving ASP problems.

In [14], the authors proposed a three-stage integrated approach with some heuristic working rules to assist the planner in generating an effective assembly sequence. In this work, the back-propagation neural network (NN) was employed to optimize the available assembly sequence.

The results show that the proposed model can facilitate assembly sequence optimization and allows the designer to recognize the contact relationship and assembly constraints of three-dimensional components.

Other approaches to this problem were presented of the basis of immune algorithm (IA) [15,16], scatter search algorithm (SSA) [17], particle swarm optimization (PSO) [15] and other heuristic methods (HM) [4,18–20]. Figure 1 shows the published research on optimization ASP from analyzed articles and conference papers from the last twenty years based on the Google Scholar database.

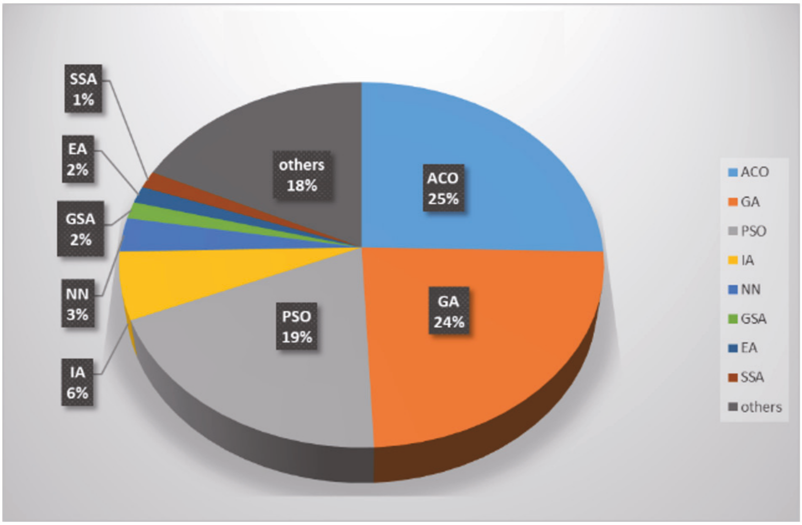


Figure 1. Frequency of the algorithms of the optimization method usage in ASP.

2.1. Assembly Sequence Assessment Criteria

The planning of assembly sequences (ASP) consists of determining the feasible and, at the same time, the most advantageous order of joining assembly units due to certain criteria. The great complexity of the task of selecting the appropriate one from among all the acceptable and, at the same time, feasible ones, due to the constraints of the structural nature of the assembly sequence, is a consequence of the large number of possible combinations. It is, therefore, necessary, in the first stage, to limit the set of possible solutions, and then to evaluate them and select the most advantageous assembly sequence. This is inextricably linked with the use of certain criteria for this purpose, due to which the discussed process can be optimized. It is very often necessary to assign appropriate priorities/weights to the criteria used for the assessment that are tailored to the specific assembly process. In the literature on the subject, the assessment and selection of the most favorable sequence can be made according to various criteria, depending on the specifics of a process in the plant where the assembly takes place. Such criteria may be: the assembly time, the number of changes in the assembly directions, the number of tool changes, the stability of the assembly states that arise, the degree of difficulty in reaching the next process state, the complexity of the assembly unit movements, the correctness of the assembly itself or even the cost-effectiveness of the process. Selected data on the criteria for evaluating assembly sequences can be obtained automatically from the electronic construction documentation or supplemented on the basis of a case-by-case analysis.

2.2. Assembly Sequence Assessment Criteria Based on DFA Rating Factors

Design For Assembly (DFA) is one of the methodologies supporting the design of the assembly process. By introducing design changes, in line with the guidelines of the DFA methodology, we can achieve, above all, shorter assembly times, by reducing unnecessary components and the assembly tasks necessary to assemble the product. DFA analysis also highlights the possible weak points of the structure and helps to create a product that does not require major changes in further phases of the product lifecycle. Thanks to the introduction of DFA to the design process, the product development team proposes improved design solutions that are characterized by better indicators, such as a simple structure, which directly simplifies the assembly operations. The benefits of using this methodology are mainly due to the systematic review of functional requirements and the replacement of groups of elements by single integrated units–assembly modules. Generally, the designer carries out the presented analyses in a series of assessment charts. Next, the designer assesses the component functionality, manufacturing processes, form and assembly characteristics using values extracted from the charts according to component properties. Thus, the designer can quantify the suitability of the design. The best known DFA methods are the Boothroyd–Dewhurst System, the Lucas DFA Methodology and the Hitachi Assembly Ability Evaluation Method [21–23]. The method proposed in this article was based on assembly sequence determination using artificial neural networks. It is related to manual assembly and is largely based on the assessment of transitions between individual assembly states using a score taken from the following DFA rating factors charts (these data are used to train the neural network):

- The stability of the assembly unit (Table 1).

The stability criterion determines the assembly sequences with the fewest unstable states. A stable unit is one that remains in the assembled position regardless of the force applied. The acting forces can be the force of gravity or the force associated with the movement of the part/assembly.

Table 1. Assembly unit stability index based on the DFA (own study based on [22,23]).

The Stability of the Assembly Unit (Based on the DFA)	
	Rating
Assembly unit/part secured after assembly	9
Assembly unit/part added but not secured	4
Additional securing of the assembly unit/part required or stability of the assembly unit/part impossible	3

- The movement and orientation index.

This is the criterion of the ability to move and orientate. Handling difficulties from the Boothroyd–Dewhurst table are rated from "easy to deliver and orientate—9" to "Requires gripping tools—0". Details of the assessment are given in Table 2.

Table 2. The criterion of the ability to move and orientate based on the DFA (own study based on [22,23]).

The Ability to Move and Orientate (Based on the DFA)	
	Rating
Easy to deliver and orientate (1-handed)	9
Delivery and orientation (2-handed)	8
Need to change orientation (1-handed)	7
Need to change orientation (2-handed)	6
Requires orientation tools	3
Requires grasping tools	2

- The ease of joining index.

The ease of connection criterion evaluates whether the part is easy to grasp and assemble in a given process state (Table 3).

Table 3. The ease of joining index (own study based on [22,23]).

The Ease of Joining Index (Based on the DFA)	
	Rating
Easy to put on (1-handed)	9
Difficulties in putting on or adjusting (1-handed)	7
Necessity of using both hands	6
Indispensable standard tool	6
Indispensable special tool	3
Required, keeping in the lower position when putting on	2
High resistance when putting on	0

- The accessibility of the joining location index (Table 4).
  The accessibility determines whether it is sufficient to secure the part. Accessibility depends on the location of the parts in the product or its subassembly.

Table 4. The accessibility of the joining location based on the DFA (own study based on [22,23]).

The Accessibility of the Joining Location (Based on the DFA)	
	Rating
Easy-to-work area, sufficient space for hands and tools	9
Limited space but the part can be installed (1-handed)	6
Limited space but the part can be installed (2-handed)	4
Difficult access	3
Blind assembly	1
Difficult blind assembly	0

3. Artificial Neural Networks in Assembly Sequence Planning Problem

An assembly sequence planning problem belongs to a class of optimization problems known as NP-complete. For such a class of problems, in order to find an optimal solution with regard to specific criteria, it is necessary to search the entire set of feasible solutions to ensure finding the optimal assembly sequence. With high complexity and the interdisciplinary nature of assembly problems, this strategy is very time-consuming and not practical in most industrial applications. For this reason, other heuristic techniques are often used in this case, allowing to find a solution that is close to the optimal one. The authors of this paper proposed a new approach to the problem of planning the assembly sequences. For this purpose, they used artificial neural networks to predict the assembly time determined by the assessment of transitions between its individual states according to the selected DFA criteria. This solution is an information-processing model inspired by a natural biological system, in which the interconnected processing elements (neurons) x_n perform an input operation that will ultimately solve the problem of assessing the assembly sequence, whose optimization criterion comes down to determining the assembly time. Neurons are associated with synapses that are assigned weights with specific values, wk. Additionally, the intercept w_0–bias is defined as the sensitivity threshold. A neuron is in an active state when the sum of the weighted input signals is greater than or equal to the sensitivity threshold w0. The vector of input data introduced into the network is multiplied by the synaptic weights according to the function of the postsynaptic potential to obtain the signal y, which is the output value of the network. Figure 2 shows the model of signal processing between neurons. The artificial neural network processes information by activating the neurons of the hidden and output layers with activation functions (transfer functions). The literature gives the highest efficiency of the logistic, exponential, linear, sine and hyperbolic tangent functions, which were used in the prediction model developed in this publication.

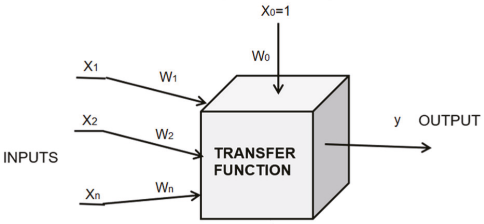


Figure 2. Schematic representation of the artificial neural network.

4. The Concept of the System Supporting the Determination of the Assembly Sequence Based on Artificial Neural Networks and DFA Rating Factors for Manual Assembly

The proposed method was based on artificial neural networks and selected DFA rating factors and aimed to assist in determining the order of manual assembly. It was assumed that, at the current stage of research, it is used in a specific mechanical production company, where the conditions of the assembly process are relatively constant for the introduced new products, which undergo the ASP process, and the assembly processes carried out in it were used to teach the network. Such a system, together with an increase in the number of analyzed assembly processes, improves the efficiency of new process time estimation. The stability of the conditions are related to a specific production company and its concerns, for example, the available machine park, production organization, process control and control, and the level of training of employees (these are constant elements for the network). As input data for the process, the previously discussed assessments of transition between individual assembly states were used based on selected DFA rating factors. The operation of the method is, therefore, aimed at estimating the time for all sequences permissible due to the constraints of a constructional nature (feasible), and thus enables the selection of the most favorable one, due to the analyzed criteria in relation to a specific assembly process.

4.1. Product Structure

The modified directed assembly state graph was adopted as the structure of the product intended for assembly. The parts in this digraph (or assembled assemblies) are marked as vertices, while the directed edges show possible sequences (paths) of joining them. It was assumed that the assembly of successive elements takes place by attaching to the n-th grade assembly a part or subassembly consisting of more parts (treated as a single assembled part). The directed edges in the digraph connecting the vertices contain information based on the assessment criteria (based on the DFA rating factors) for a specific transition between assembly states. The described digraph can be generated automatically on the basis of a CAD assembly drawing. The basis for carrying out the ASP, due to the specific criteria for a specific assembly process, is to obtain a set of all allowed and feasible assembly sequences. The matrix notation (e.g., in the form of a matrix of assembly states or a matrix of an assembly graph) of assembly units enables the determination of all variants of the order of joining assembly units using an appropriate algorithm (this procedure is not discussed in this paper, but comes down to finding all paths in the digraph leading from the vertex starting xs, constituting the base part, to the end vertex x_e, i.e., the last state of the already assembled product—$x_s, \ldots, x_e$) [23–27].

4.2. Assumptions of the Neural Network

In order to develop a predictive model for the evaluation of the assembly sequence, a set of data to teach the neural network was prepared, including input and output data. The input data were related to the DFA assessment criteria, which were divided into four groups: stability, the ability to change the orientation of the assembly unit, the ease of joining parts, and space availability during the connection process. The assembly time was included in the group of output data. The collection of an appropriate number of training data and the links between them was made. The numerical data entered into the network were normalized using the min–max function to values ranging from 0 to 1, according to the linear transformation, in order to ensure their uniformity and the compatibility between the variables in the process of signal processing by the neural network. For the assembly time prediction task, the network model with the best efficiency was selected, which was obtained by empirically testing its various parameters: the number of hidden neurons, the activation function and the network-learning algorithm. The neurons in the hidden layers process the signals from the input neurons and transform them into intermediate data that are then passed on to the output neuron. Hidden neurons allow for modeling complex relationships between data, and their number should be selected so that the structure of the network is not so extensive, while ensuring correct data processing. The network model takes into account the presence of an additional neuron with a value of +1, which is a generator of an artificial neuron causing neuron polarization and, as a result, improving the stability of the network in the process of learning it. The functions of neuron activation in the hidden and output layers, which were used in modeling the structure of the neural network, are shown in Table 5.

Table 5. Functions of neuron activation.

Function	Formula	Description
Linear	$f(x) = x$	Linear transmission of the neuron value to the network output, which takes values in the range from $-\infty$ to ∞
Logistics (sigmoidal)	$f(x) = \frac{1}{1-e^{-1}}$	A sigmoid curve with values greater than 0 and less than 1
Exponential	$f(x) = e^{-x}$	An exponential curve with values from 0 to ∞
Hyperbolic tangent	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	A hyperbolic tangent curve with values from -1 to +1
Sine	$f(x) = \sin(x)$	A sine curve with values from 0 to 1

The following algorithms were used to train the neural network: steepest descent, gradient scaling and Broyden–Fletcher–Goldfarb–Shanno. The principle of their operation boils down to minimizing the error function as a result of an iterative change of weights describing the neurons. The steepest descent method consists of finding the minimum of the error function in a given search direction until it turns out to be tangential to a certain line defining the constant value of the objective function. Successive directions of the search for the optimum of a function are orthogonal to the previous ones. The gradient scaling method does not require so many directions of searching for the minimum function. It consists of finding the correct direction on the multidimensional parabolic-shaped error surface on which the straight line is drawn and the minimum of the function for all points on the straight line is determined. Finding a minimum along a given direction results in determining a new search direction from that point. Repeating the process continues until the minimum of the function is found, according to the implementation of the constant shift towards decreasing values of the error function. The second derivative of the function in the following steps of the algorithm is zero. Maintaining this value is possible due to the existence of directions coupled with the previously selected directions. The Broyden–Fletcher–Goldfarb–Shanno method consists of changing the weights of individual neurons after each iteration of the algorithm as a result of taking into account the average error gradient. The search for the minimum error function is carried out by the steepest slope method, and then by the inverse estimation of the matrix of the second-order partial derivatives (Hessian). The division of the examples introduced into the network (input and output data) into training, testing and verification subsets also influences the achievement of the expected network results. Sufficient data should be provided for each of these subsets: (1) To understand the relationship between the data and learn to adapt to changing conditions. Cases from this set affect the change of other network parameters, e.g., the weight values assigned to individual neurons. (2) To check the results obtained by the network and the ability to generalize. Cases from this set are not used to modify network parameters. (3) To verify the network results on the basis of quantitative or qualitative datasets that were not used before. The selection of the best parameters of the artificial neural network was made on the basis of the correlation coefficient determining the effectiveness of the learning and testing process as well as the interpretation of the standard error function—the sum of the squared differences between the set values and those obtained in the output neuron:

$$SOS = \sum_{i=1}^{n} (y_i - y_i^*) \tag{1}$$

where n is the number of datasets teaching the neural network; y_i is the expected value in the output neuron; and $y_i{}^*$ is the actual value in the output neuron.

5. Example of Determining the Assembly Sequence

An example of determining the assembly sequence using artificial neural networks was made for a tractor door, consisting of 8 main assembly units such as: door–welded structure (part no. 1), lock (part no. 2), cassette lock (part no. 3), door reinforcement bar with passenger's handle (part no. 4), lock cover (part no. 5), door seal (part no. 6), lower glass (part no. 7) and upper glass (part no. 8). Next, the basic part from which the assembly begins was assumed (door–welded structure, part no. 1). It was assumed that the assembly of the next units takes place by adding another assembly unit to the assembly state of the n-th stage. Then, the digraph and state matrix of the structural limitations of the assembly were built (shown in Figure 3). Then, using the selected graph-searching algorithm (Dijkstry), all feasible (due to the constraints of the structural nature) assembly sequences were determined, which constitute the basis for further analysis.

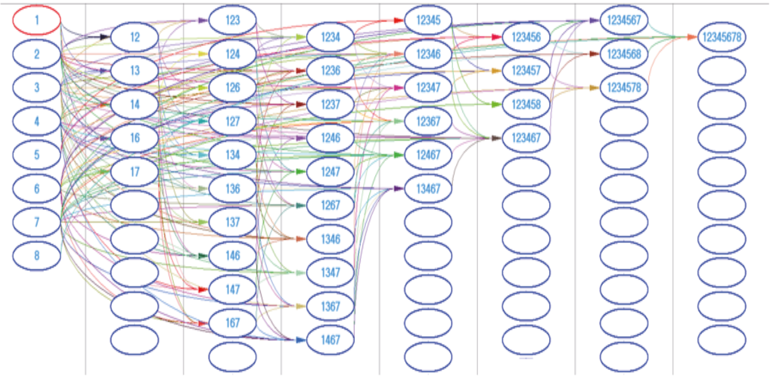


Figure 3. Schematic representation of the artificial neural network.

The neural network model for planning the assembly sequence was selected on the basis of a multiple sampling algorithm that randomly presents 20,000 network variants with variable parameters:

- The number of neurons in the hidden layer (from 1 to 15);
- Values of the weights of individual neurons in the network (from −1 to +1);
- Activation functions (linear, sigmoidal, exponential, hyperbolic tangent and sine);
- Network learning algorithms (steepest gradient, scaled conjugate gradient and Broyden–Fletcher–Goldfarb–Shanno).

The constant parameters of the neural network are:

- The number of input neurons (4)—the groups of DFA assessment criteria (stability, the possibility of changing the orientation of the assembly unit, ease of joining parts and space availability during the connection process);
- The number of output neurons (1)—assembly time;
- The division of the dataset entered into the neural network—70% training data, 15% testing data and 15% verification data;
- The type of predictive model—regression;
- The topography of the neural network—multilayer perceptron.

The neural network with the best parameters, confirmed by the highest correlation coefficient in the group of test and verification data, is given in Table 6. The model of the MLP 4–2–1 network structure is shown in Figure 4. The network selected for subsequent research consists of four input neurons, two hidden neurons and one output neuron. The Broyden–Fletcher–Goldfarb–Shanno algorithm was used for the network learning process, and the neurons in the hidden layer were activated with the exponential function and the output with the sine function. These network parameters made it possible to observe a strong correlation between the data, with the highest coefficient in the verification group $R^2 > 0.9$ and the smallest SOS error < 0.1. This group is the most important, concerns data not included in the earlier stages of the analysis and presents the results of the predictions most reliably.

Table 6. Parameters of the MLP 4–2–1 neural network selected for the prediction task.

Effectiveness (Learning)	Effectiveness (Testing)	Effectiveness (Verification)	SOS Error (Learning)	SOS Error (Testing)	SOS Error (Verification)	The Learning Algorithm	Activation (Hidden Neurons)	Activation (Output Neurons)
0.589	0.887	0.976	0.018	0.062	0.010	BFGS	exponential	sinus

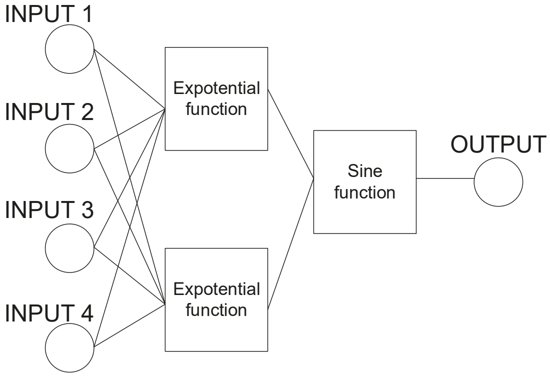


Figure 4. The structure of the MLP 4–2–1artificial neural network.

The minimum of the error function was found in the successive network training cycles. For the MLP 4–2–1 network, the optimum of the function was found in the 12th epoch in which the stabilization of the error value was noticeable. Figure 5 shows a graph of the dependence of the network error value on the number of iterations of the training algorithm.

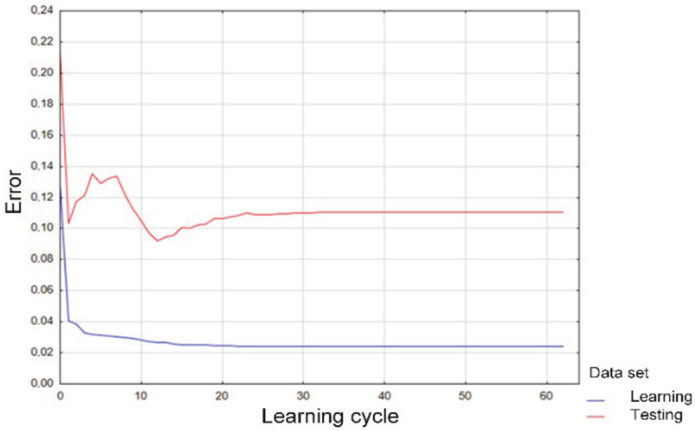


Figure 5. Graph of the error function in the successive epochs of the training algorithm.

The application of the proposed solution requires the verification of the actual data obtained at the network output with the expected data, which are summarized in Table 7, while the linear regression charts for these data are shown in Figure 6.

Table 7. Comparison of the prediction results with the expected data (randomly selected examples of the results).

Expected Network Output	Real Network Output
0.44	0.47
0.53	0.51
0.58	0.58
0.46	0.45
0.57	0.56
0.48	0.49
0.58	0.56
0.59	0.57

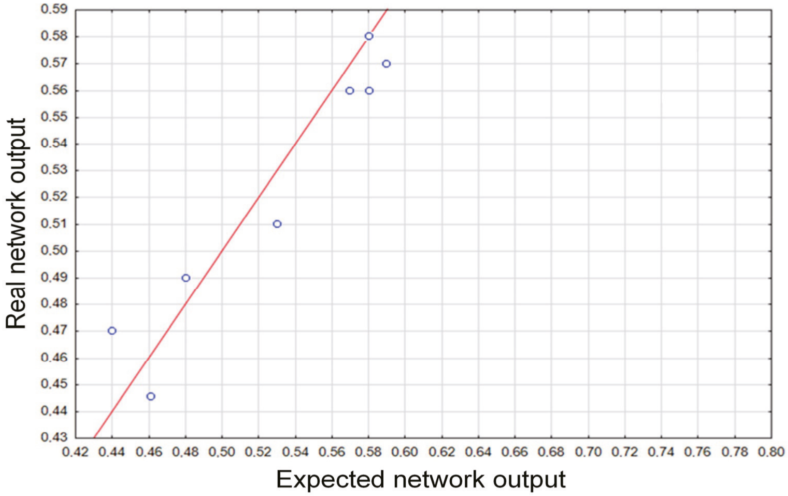


Figure 6. Linear regression chart for the normalized expected and actual values (randomly selected examples of the results).

6. Conclusions

The developed model of the artificial neural network supporting the assembly sequence planning was positively verified using data not included in the training algorithm. The prediction results are characterized by good correlation coefficients $R^2 > 0.9$ for the group of verification data and an SOS error < 0.1. The predictive model presented in this publication is the beginning of work on the development of a universal tool for assessing the assembly sequence of various products, and thus obtaining a finished product in the shortest possible time. The use of the part DFA methodology to evaluate assembly sequences, which are the basis for network learning, is a novelty in this research area.

The effectiveness of the method depends mainly on the number of cases teaching the neural network that are able to generalize the knowledge and the neural network. Thus, a network constraint may be the presence of a large number of errors when predicting the assembly time on the base of DFA factors for other products.

The simulation results also suggest that the proposed neural predictor could be used as a predictor for assembly sequence planning system. Further research will be aimed at extending the learning dataset and verifying the assumptions of the network model made for other products in a specific industrial plant.

The aim of the authors is to develop the conducted research and verify the operation of the network on a wide range of products.

Author Contributions: Conceptualization, M.S. (Marcin Suszyński); methodology, M.S. (Marcin Suszyński) and K.P.; software, M.S. (Marcin Suszyński) and K.P.; validation, M.S. (Marcin Suszyński) and K.P.; formal analysis, M.S. (Marcin Suszyński) and K.P.; investigation, M.S. (Marcin Suszyński) and K.P.; resources, M.S. (Marcin Suszyński) and K.P.; data curation, M.S. (Marcin Suszyński); writing—original draft preparation, M.S. (Marcin Suszyński) and K.P.; writing—review and editing, M.S. (Marcin Suszyński) and K.P.; visualization, M.S. (Marcin Suszyński) and K.P.; supervision, M.S. (Marcin Suszyński); project administration, M.S. (Marcin Suszyński) and K.P.; funding acquisition; M.S. (Martin Svoboda) and V.Č.; validation, M.S. (Martin Svoboda) and V.Č.; review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Higher Education of Poland (No. 0614/SBAD/1547).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Orosz, T.; Rassõlkin, A.; Kallaste, A.; Arsénio, P.; Pánek, D.; Kaska, J.; Karban, P. Robust Design Optimization and Emerging Technologies for Electrical Machines: Challenges and Open Problems. *Appl. Sci.* **2020**, *10*, 6653. [CrossRef]
2. Selvanayaki, K.; Kalugasalam, P. Intelligent brain tumor tissue segmentation from magnetic resonance image (mri) using meta heuristic algorithms. *J. Glob. Res. Comput. Sci.* **2013**, *4*, 13–20.
3. Ibrahim, I.; Ibrahim, Z.; Ahmad, H.; Jusof, M.F.M.; Yusof, Z.M.; Nawawi, S.W.; Mubin, M. An assembly sequence planning approach with a rule-based multi-state gravitational search algorithm. *Int. J. Adv. Manuf. Technol.* **2015**, *79*, 1363–1376. [CrossRef]
4. Abdullah, M.A.; Rashid, M.F.F.A.; Ghazalli, Z. Optimization of Assembly Sequence Planning Using Soft Computing Approaches: A Review. *Arch. Comput. Methods Eng.* **2019**, *26*, 461–474. [CrossRef]
5. Chen, S.-F.; Liu, Y.-J. An adaptive genetic assembly-sequence planner. *Int. J. Comput. Integr. Manuf.* **2001**, *14*, 489–500. [CrossRef]
6. Wang, D.; Shao, X.; Liu, S. Assembly sequence planning for reflector panels based on genetic algorithm and ant colony optimization. *Int. J. Adv. Manuf. Technol.* **2017**, *91*, 987–997. [CrossRef]
7. Li, X.; Shang, J.; Cao, Y. An efficient method of automatic assembly sequence planning for aerospace industry based on genetic algorithm. *Int. J. Adv. Manuf. Technol.* **2017**, *90*, 1307–1315.
8. Shan, H.; Zhou, S.; Sun, Z. Research on assembly sequence planning based on genetic simulated annealing algorithm and ant colony optimization algorithm. *Assem. Autom.* **2009**, *29*, 249–256. [CrossRef]
9. Murali, G.B.; Deepak, B.B.V.L.; Bahubalendruni, M.V.A.R.; Biswal, B.B. Optimal Assembly Sequence Planning Towards Design for Assembly Using Simulated Annealing Technique. In *Research into Design for Communities*; Springer: Singapore, 2017; Volume 1, pp. 397–407.
10. Kaur, S.; Agarwal, P.; Rana, R.S. Ant colony optimization: A technique used for image processing. *Int. J. Comput. Sci. Technol.* **2011**, *2*, 173–175.
11. Wang, J.F.; Liu, J.H.; Hong, Y.F. A novel ant colony algorithm for assembly sequence planning. *Int. J. Adv. Manuf. Technol.* **2005**, *25*, 1137–1143. [CrossRef]
12. Shi, S.C.; Li, R.; Fu, Y.L.; Ma, Y.L. Assembly sequence planning based on improved ant colony algorithm. *Comput. Integr. Manuf. Syst.* **2010**, *16*, 1189–1194.
13. Zeng, C.; Gu, T.; Zhong, Y.; Cai, G. A Multi-Agent Evolutionary algorIthm for Connector-Based Assembly Sequence Planning. *Procedia Eng.* **2011**, *15*, 3689–3693. [CrossRef]
14. Chen, W.; Tai, P.; Deng, W.; Hsieha, L. A three-stage integrated approach for assembly sequence planning using neural networks. *Expert Syst. Appl.* **2008**, *34*, 1777–1786. [CrossRef]
15. Zhang, H.; Liu, H.; Li, L. Research on a kind of assembly sequence planning based on immune algorithm and particle swarm optimization algorithm. *Int. J. Adv. Manuf. Technol.* **2013**, *71*, 795–808. [CrossRef]
16. Biswal, B.B.; Pattanayak, S.K.; Mohapatra, R.N.; Parida, P.K.; Jha, P. Generation of optimized robotic assembly sequence using immune based technique. In Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Houston, TX, USA, 9–15 November 2012.
17. Martı', R.; Laguna, M.; Glover, F. Principles of scatter search. *Eur. J. Oper. Res.* **2006**, *169*, 359–372. [CrossRef]
18. Guo, J.; Sun, Z.; Tang, H.; Yin, L.; Zhang, Z. Improved cat swarm optimization algorithm for assembly sequence planning. *Open Autom. Control Syst. J.* **2015**, *7*, 792–799. [CrossRef]
19. Li, X.; Qin, K.; Zeng, B.; Gao, L.; Su, J. Assembly sequence planning based on an improved harmony search algorithm. *Int. J. Adv. Manuf. Technol.* **2015**, *84*, 2367–2380. [CrossRef]
20. Ghandi, S.S.; Masehian, E. A breakout local search (BLS) method for solving the assembly sequence planning problem. *Eng. Appl. Artif. Intell.* **2015**, *39*, 245–266. [CrossRef]
21. Sąsiadek, M.; Jakubowski, J. Product Development in Virtual Computer Integrated Engineering Systems using DfM and DfA Principles-part II. *ICMD* **2016**, *2016*. Available online: https://www.icmd2016.zcu.cz/public/conferences/1/schedConfs/1/program-en_US.pdf (accessed on 10 April 2022).
22. Shetty, D.; Ali, A. A new design tool for DFA/DFD based on rating factors. *Assem. Autom.* **2015**, *35*, 0144–5154. [CrossRef]

23. Suszyński, M.; Ciszak, O. Selection of Assembly Sequence for Manual Assembly Based on DFA Rating Factors. In *Advances in Manufacturing II. Volume 2—Production Engineering and Management*; Hamrol, A., Kujawińska, A., Francisco, M., Barraza, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; pp. 45–57.
24. Suszyński, M.; Żurek, J. Computer aided assembly sequence generation. *Manag. Prod. Eng. Rev.* **2015**, *6*, 83–87.
25. Suszyński, M.; Żurek, J.; Legutko, S. Modelling of assembly sequences using hypergraph and directed graph. *Teh. Vjesn. Tech. Gaz.* **2014**, *21*, 1229–1233.
26. Suszynski, M.; Peta, K. Assembly sequence planning using artificial neural networks for mechanical parts based on selected criteria. *Appl. Sci.* **2021**, *11*, 10414. [CrossRef]
27. Butlewski, M.; Czernecka, W.; Pajzert, A.; Radziejewska, M.; Suszyński, M.; Feniser, C. Ergonomic criteria in the optimization of assembly processes. In *Proceedings of the 6th RMEE2018—Performance Management or Management Performance*; Publishing House: Cluj-Napoca, Romania, 2018; pp. 424–431.

applied sciences

MDPI

Article

Sensitive Ant Algorithm for Edge Detection in Medical Images

Cristina Ticala *,†, **Camelia-M. Pintea** † **and Oliviu Matei** †

Technical University of Cluj-Napoca, North University Center of Baia Mare, 400114 Cluj-Napoca, Romania; dr.camelia.pintea@ieee.org (C.-M.P.); oliviu.matei@holisun.com (O.M.)
* Correspondence: cristina.ticala@cunbm.utcluj.ro
† These authors contributed equally to this work.

Abstract: Nowadays, reliable medical diagnostics from computed tomography (CT) and X-rays can be obtained by using a large number of image edge detection methods. One technique with a high potential to improve the edge detection of images is ant colony optimization (ACO). In order to increase both the quality and the stability of image edge detection, a vector called pheromone sensitivity level, PSL, was used within ACO. Each ant in the algorithm has one assigned element from PSL, representing the ant's sensibility to the artificial pheromone. A matrix of artificial pheromone with the edge information of the image is built during the process. Demi-contractions in terms of the mathematical admissible perturbation are also used in order to obtain feasible results. In order to enhance the edge results, post-processing with the DeNoise convolutional neural network (DnCNN) was performed. When compared with Canny edge detection and similar techniques, the sensitive ACO model was found to obtain overall better results for the tested medical images; it outperformed the Canny edge detector by 37.76%.

Keywords: medical image edge detection; image processing; fixed point; Krasnoselskij iteration; admissible perturbation; ant colony optimization

Citation: Ticala, C.; Pintea, C.-M.; Matei, O. Sensitive Ant Algorithm for Edge Detection in Medical Images. *Appl. Sci.* **2021**, *11*, 11303. https://doi.org/10.3390/app112311303

Academic Editor: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Received: 1 November 2021
Accepted: 22 November 2021
Published: 29 November 2021

1. Introduction

Today, medicine is interconnected with technology. Human injuries caused by accidents or other similar events can be detected and correctly diagnosed by using tomography or X-rays. In medical image processing, edge detection has a major role. In order to obtain accurate medical diagnoses, the best computing models are involved. As swarm intelligence has a huge impact nowadays in solving complex problems, the current work uses a particular swarm method, ant colony optimization (ACO) [1], to solve the edge detection problem.

Learning is one of the most efficient artificial intelligence capabilities; in [2], learning with PDE-based CNNs and dense nets for the purpose of detecting COVID-19, pneumonia, and tuberculosis from chest X-ray images was studied. In the same context, automatic COVID-19 detection from chest X-ray and CT-scan images was proposed [3] within a new meta-heuristic feature selection using an optimized convolutional neural network [4].

"A meta-heuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions" [5]. In general, the quality of heuristics solutions, including bio-inspired methods such as ACO, is given by appropriate probabilistic assumptions [6].

One of the most recent works related to medical image edge detection with ant colony optimization shows the efficiency of a gradient-based ant spread modification to ACO for retinal blood vessel edge detection [7]. In [8], a new image filtering method is introduced for the problem of edge extraction for some targets according to the top-down information based on the image perspective effect; the authors assign scale and orientation, in a hard manner, in order to enhance a local edge detection.

Ant colony optimization is one of the most successful metaheuristics used within complex combinatorial optimization problems, as, for example, in scheduling, transportation and test case prioritization problems [9–12]. Pintea and Pop introduced and showed the benefits of agent sensitivity, including sensitive robots, in security-related problems [13,14], such as a denial jamming attack on sensor networks. As a reference, the current work makes use of the state-of-the-art Canny [15] edge detection technique. Recently, the Canny operator was used in [16] during a symmetrical difference kernel SAR image edge detection process.

The current paper introduces a version of ant colony method with a specific feature called pheromone sensitivity level, PSL, for solving the medical edge detection problem. The artificial ants are endowed with different levels of artificial pheromone sensitivity; thus, the agents have different reactions in a dynamic environment. Here, the new algorithm is applied to the image edge problem and requires a heuristic value computed with two admissible perturbation operators applied to a demicontractive mapping.

Pintea and Ticala proposed the first related theoretical approach in [17]; a step forward was made in [18]. It includes more tests for both ant colony versions of medical image edge detection and a comparison of these techniques; details, including the efficiency of the new parameters and the use of some demicontractive operators, are presented.

The current work's content is as follows:

- Sensitive ant colony optimization (SACO) for medical image edge detection is introduced to improve the analysis of CT and X-ray images.
- Image pre-processing is done.
- The use of several demi-contractive operators is employed to check their behavior for both ACO and SACO.
- Postprocessing, including the use of the DeNoise convolutional neural network (DnCNN), is done.
- A comparison between ACO, SACO and several state-of-art methods to ensure the validity of the sensitive approach on a CT and X-ray image dataset is made.

The next section includes the present work's prerequisites with mathematical support, the edge detecting problem and the sensitive ant colony optimization (SACO) method. The numerical tests and bio-inspired methods results follows in Section 3. The comparison of methods, the operators' behavior and the representation of medical image results are discussed in Section 4. Future work and arguments regarding the benefits of ACO and SACO for medical images conclude the present study.

2. Prerequisites

Math Operators. At first, a short introduction into the mathematical part of the work is presented; this is, mainly based on Rus [19] who introduced the theory of admissible perturbations of an operator. The admissible perturbation operator was also studied in [20].

Demicontractive operators. As already stated in our previous work [18], a *demicontractive operator* (T) is defined by C, a subset of $\mathbb{R}$ (domains and co-domains). For an existing *contraction coefficient* ($k < 1$), each fixed point (p) of the demicontractive operator and all numbers ($x \in C$) the inequality (1) is true.

$$\|Tx - p\|^2 \leq \|x - p\|^2 + k\|x - Tx\|^2. \tag{1}$$

For a nonempty set (X) and an admissible mapping, ($G : X \times X \rightarrow X$), the following statements are true: for all $x \in X$ G(x, x) = x and $G(x, y) = x$ implies $y = x$ [19].

The admissible perturbation of the operator f ($f : X \rightarrow X$) [19] is the admissible mapping $f_G : X \rightarrow X$ ($f_G(x) := G(x, f(x))$).

Krasnoselskij operator. The *Krasnoselskij algorithm* [19], corresponding to an admissible mapping ($G : X \times X \rightarrow X$) of an nonlinear operator ($f : X \rightarrow X$), is defined as an iterative algorithm $\{x_n\}_{n\in\mathbb{N}}$ with $x_0 \in X$ and $x_{n+1} = G(x_n, f(x_n))$, where $n \geq 0$.

For further details and examples, see [18,20,21].

The χ operator. The χ operator $\chi : \mathbb{R} \times \mathbb{R} \longrightarrow [0, 1)$ is defined as:

$$\chi(x, y) = \frac{x^2 \cdot y^2}{(1 + x^2) \cdot (1 + y^2)}, \tag{2}$$

where $y : \mathbb{R} \to \mathbb{R}$, $y(x) = \frac{2}{3} x \sin \frac{1}{x}$, if $x \neq 0$.

Particular Math Functions. The operators further used as $f(\cdot)$ in the considered methods (see (8)) are as follows:

$$\text{Sin:} \qquad f(x) = \sin\left(\frac{\pi x}{2\lambda}\right), \text{ if } 0 \leq x \leq \lambda; \tag{3}$$

$$\text{KH:} \qquad f(x) = (1 - \lambda) \cdot x + \lambda \cdot \frac{2}{3} x \sin \frac{1}{x} \text{ if } x \neq 0; \tag{4}$$

$$\text{Chi:} \qquad f(x) = (1 - \chi(x, y(x))) \cdot x + \chi(x, y(x)) \cdot y(x) \text{ if } x \neq 0; \tag{5}$$

The functions (3)–(5) are zero in all other possible cases. The λ parameter from Equations (3) and (4) adjusts the operators used as test functions in [22]. In [23], the authors used two admissible perturbation operators for computing the heuristic value required within the ACO algorithm.

In the present article, where admissible perturbations of demicontractive mappings are utilized as test functions, and the PSL vector is utilized for each ant, a sensitivity to the artificial pheromone is introduced using a specific coefficient influenced by the image's intensity values for the edge detection problem.

The ants have different roles in edge extraction: some agents are explorers and others are exploiters; these roles are exchanged as the PSL vector updates during processes. The obtained results for CT and X-ray medical images and the comparison among the results using the proposed operators are made in Section 4.

3. Problem and Methods

3.1. Medical Image Edge Detection Problem

The problem to solve is the edge detection problem. The current work improves solutions of particular medical images due to their complex edges based on X-rays and tomographic images.

Image edge detection involves the detection of discontinuities in brightness while processing the image in order to find the boundaries of objects.

3.2. Sensitive Ant Colony Optimization Method

The method used is an improved version of *Ant Colony Optimization (ACO)* [1] called the *Sensitive Ant Colony Optimization (SACO)* [24,25]. The ant colony optimization sensitive approach for medical image edge detection is further presented. There is considered a colony of K ants engaged in a search within the graph space $\mathcal{X}$, with $M_1 \times M_2$ nodes.

SACO as well as ACO use artificial ants to move in a 2D image in order to build the pheromone matrix; each matrix element represents the edge information for every pixel in the image.

In general, ACO and its versions builds a solution with the use of artificial ants; these agents search for the best path in a given space by depositing artificial pheromones [1,23]. These pheromone trails are updated during the search process.

The ACO and SACO general scheme includes an initialization process followed by N construction steps while creating and updating pheromone matrix, and, finally, performing the decision process to determine a beneficial solution.

Ant colony optimization for edge detection

```
Initialize ACO parameters
Schedule activities
  Construct ant solutions
  Update pheromone
  Edge detection
End scheduled activities
```

Initialization process: In particular, for image edge detection with ACO and SACO during the initialization process, the entire ant colony (K ants) places ants randomly on the image matrix. Each image pixel is considered to be a node in a graph. Each initial pheromone matrix $\tau_{(0)}$ has a constant τ_{init} value. A constant value L defines the number of moves during the construction process.

For SACO in particular, each PSL vector component is initialized with 1, starting as the original ACO (see Figure 1).

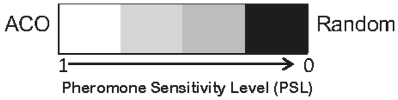


Figure 1. Symbolic illustration of the sensitive ant model showing the ants' probability variation within the unit interval from the ACO probability when $PSL = 1$ to a random walk probability when $PSL = 0$.

Construction phase: at the n-th construction step, a randomly chosen ant will move from node i to j according to the transition probability in (6) for L steps.

$$p_{ij}^{n} = \frac{\left(\tau_{ij}^{(n-1)}\right)^{\alpha}(\eta_{ij})^{\beta}}{\sum_{j\in\Omega_i}\left(\tau_{ij}^{(n-1)}\right)^{\alpha}(\eta_{ij})^{\beta}}, \qquad \text{if } j \in \Omega_i, \tag{6}$$

where τ_{IJ} is the pheromone value on (i, j); η_{IJ} the heuristic value connecting nodes i and j the same for all n construction steps; α and β are the weighting factors for the pheromone and the heuristic; and Ω_i includes the neighborhood nodes of node i.

The overall eight-connectivity neighborhood for each pixel $I_{i,j}$ within the local configuration at the $I_{i,j}$ pixel, $cI_{i,j}$, for computing the variation value $V_c(I_{i,j})$ defined by (8) is illustrated in [18].

Here, we propose the computation of $\eta_{i,j}$ according to the local statistic of the pixel (i, j) (Equation (7)).

$$\eta_{i,j} = \frac{1}{Z} \cdot V_c(I_{i,j}), \tag{7}$$

where $Z = \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} V_c(I_{i,j})$ is a normalization factor, $I_{i,j}$ is the intensity value of the image pixel (i, j); and the function $V_c(I_{i,j})$ processes the "clique" $cI_{i,j}$ [22].

The $V_c(I_{i,j})$ value at pixel $I_{i,j}$ is influenced by the image's intensity values for $cI_{i,j}$, and its value is [22]:

$$\begin{aligned} V_c(I_{i,j}) = \quad & f\Big(\left|I_{i-2,j-1} - I_{i+2,j+1}\right| + \left|I_{i-2,j+1} - I_{i+2,j-1}\right| \\ & +\left|I_{i-1,j-2} - I_{i+1,j+2}\right| + \left|I_{i-1,j-1} - I_{i+1,j+1}\right| \\ & +\left|I_{i-1,j} - I_{i+1,j}\right| + \left|I_{i-1,j+1} - I_{i+1,j-1}\right| \\ & +\left|I_{i-1,j+2} - I_{i+1,j-2}\right| + \left|I_{i,j-1} - I_{i,j+1}\right|\Big). \end{aligned} \tag{8}$$

In order to validate an edge within the solution, a decision is made for each image pixel by applying a threshold T (see [26]) to the final pheromone matrix $\tau^{(N)}$.

The artificial pheromone matrix values are updated both locally and globally.

Locally update the pheromone matrix τ. The local pheromone matrix update is made after each ant moves within each construction step [22].

$$\text{Local update:} \quad \tau_{ij}^{(n)} = \tau_{ij}^{n-1} \cdot (1-\rho) + \rho \cdot \Delta_{ij}. \tag{9}$$

Notations: ρ is the pheromone evaporation rate, and Δ_{ij} is the artificial pheromone laid on edge (ij).

Globally update the best tour's PSL vector and pheromone matrix τ. The global update occurs after all ants finish all construction steps. Now, the PSL vector recording the pheromone sensitivity level for each ant is also updated according to a specified linear formula based on [24,25]; for this particular problem, the Equation (10) PSL update influenced by the image's intensity values is used.

$$PSL = ((1-\rho) * PSL + \rho * \Delta_{ij} * v(I_{ij})) * \Delta_{ij} + PSL * |1 - \Delta_{ij}|. \tag{10}$$

Furthermore, the best tour is a user defined criterion; it can be the best tour found in the current construction step, or the best tour from the start of the ACO algorithm, or a combination of these two.

For ACO, global update of the pheromone matrix [1] is performed as in Equation (11).

$$\text{Global update ACO:} \quad \tau^{(n)} = (1-\psi) \cdot \tau^{(n-1)} + \psi \cdot \tau^{(0)}, \tag{11}$$

where ψ is the pheromone decay rate.

For *SACO*, the global update is based on its sensitivity feature (Equation (12)).

$$\text{Global update SACO:} \qquad \tau^{(n)} = \max_{k=1:K} PSL(k) \cdot \tau^{(n-1)}. \tag{12}$$

The problem solution is obtained after reaching a stopping criteria, such as, for example, a maximal number of iterations.

4. Experiments and Discussions

The numerical experiments were carried out using Matlab on an AMD Rysen 5 2500U, 2GHz. The software is a version of the image edge detection using Ant Colony Optimization version 1.2.0.0. from MATLAB Central File Exchange [27]. The MATLAB implementation [28] of the Canny edge detection algorithm is based on [15].The software makes use of two thresholds in order to detect strong and weak edges; the weak edges are provided in the solution only if they are connected to the strongest ones: "a high threshold for low edge sensitivity and a low threshold for high edge sensitivity", as is specified in the software documentation [28]. In order to convert a gray-scale input image to a binary image, thresholding is used.

Data set. A dataset of medical images, free of copyright, was used for these experiments: *Brain CT* (could be provided by request from the authors), *Hand X-ray* [29], (reduced resolution from 225 × 225 to 128 × 128) and *Head CT* [30]. Several details are included in the Github page (Representation of results available at https://github.com/cristina-ticala/Sensitive_ACO; accessed on October 2021).

Filtering. In order to filter the medical images, the De-Noise convolutional neural network (DnCNN) was used in the present study, as well as in our previous related work [18]. The Image Processing Toolbox and Deep Learning Toolbox from Matlab [31] were used.

Parameters. Most of the parametric numbers are from [22]. In our previous work [18], we tested several parameters; in the present study, we used the best of them.

Image-related parameters: The image dimension influences and gives ACO and SACO a number of artificial ants $K = \lfloor \sqrt{M1 \times M2} \rfloor$, where $\lfloor$ and $\rfloor$ are the left and right rounded values to the nearest integers less than or equal to x; e.g., for a 128 × 128 image resolution, the number of ants is considered 128.

Iterations related parameters: In [18] just 30,000 iterations for $L = 100$ steps were considered; here, we tested a smaller (1200 iterations for $L = 4$ steps) and a higher number of steps $L = 1000$ (300,000 iterations). An ant makes 300 moves at each step; e.g., for 128 ants (e.g. image resolution: 128 × 128), 38,400 moves are made during each step. Therefore, for $L = 4$, it is a total of 153,600 moves, 3,840,000 moves for $L = 100$ and a total of 38,400,000 ants' moves for $L = 1000$ steps.

Connectivity-related parameters: The connectivity neighborhood parameter $\Sigma = 8$ is based on the ants' movement range (Equation (6)).

Pheromone trail parameters: the value of each matrix component $\tau_{init} = 0.0001$; the weighting factors of pheromone information $\alpha = 1$ and of heuristic information $\beta = 0.1$ (Equation (6)); the evaporation rate, ρ, is 0.1 (Equation (9)) and the value of the pheromone decay coefficient ψ is 0.001 (Equation (12)).

Other parameters: The adjusting factor λ of the functions (Equations (3)–(5)) is 10. The tolerance parameter ($\varepsilon = 0.1$) is used in the decision process. The stopping criterion is given by the maximal number of steps (L) set by the user.

Comparison: Beside the Canny algorithm, the Roberts, the Sobel and Prewitt edge algorithms were also used for comparison; the last two methods compute the horizontal and the vertical gradient of an image by using two orthogonal filter kernels, and after filtering, they compute the gradient magnitude and apply a threshold in order to find the regions of the image corresponding to the edges. Furthermore, the Roberts algorithm detects image edges at angles of 45 degrees and/ or 135 degrees from horizontal [32].

Table 1. The best number of correctly identified number of pixels, standardized using the overall average and standard deviation for all considered medical images, with every considered operator on all considered algorithms results for sensitive (SACO) and original ACO with DnCNN.

	Head CT		**Brain CT**		**Hand X-ray**	
	ACO	**SACO**	**ACO**	**SACO**	**ACO**	**SACO**
			1200 iterations			
Sin	0.2694	0.4137	−0.1265	0.0111	0.0346	0.0547
KH	−0.3379	−0.2406	−0.1366	0.0648	−0.6616	−0.5778
Chi	−0.3261	−0.3127	−0.1550	−0.1533	−0.7975	−0.6549
			30,000 iterations			
Sin	0.4875	0.4154	−0.0862	0.0815	0.0312	0.0547
KH	−0.2842	−0.2355	−0.0342	0.0614	−0.6214	−0.5627
Chi	−0.3798	−0.3127	−0.2372	−0.1500	−0.7036	−0.6549
			300,000 iterations			
Sin	0.4322	0.4154	−0.0980	0.0765	−0.0124	0.0547
KH	−0.2691	−0.2355	−0.1080	0.0614	−0.6080	−0.5627
Chi	−0.3882	−0.3127	−0.2221	−0.1500	−0.7841	−0.6549

Table 2. The best number of the correctly identified number of pixels, standardized using the overall average and standard deviation for all considered medical images, with every considered operator on all considered algorithms results for Canny edge detection [15], as well as the Prewitt, Sobel, and Roberts methods [32].

	Head CT	**Brain CT**	**X-ray**
Canny	−1.4166	−1.4803	−1.2857
Prewitt	−2.4567	−2.6865	−2.7721
Sobel	−2.4751	−2.5926	−2.7486
Roberts	−3.0606	−2.8878	−2.9130

Running time. The average running time was around 4500 seconds for both ACO and SACO with the presented parameters on the utilized computer.

Table 1 shows the best, maximal results of the number of correctly identified pixels standardized using the overall average, $Avg = 2107.030303$, and standard deviation, $StdDev = 563.50$, for all considered medical images, and operators on the sensitive ant colony method (SACO) and ACO denoised with DnCNN. Table 2 shows the Canny [15], Prewitt, Sobel and Roberts methods results and Table 3 illustrates the original ACO and SACO results before post-processing with DnCNN.

Table 3. The best number of correctly identified number of pixels, standardized using the overall average and standard deviation for all considered medical images, with every considered operator on all considered algorithms; results for sensitive (SACO) and original ACO methods.

	Head CT		**Brain CT**		**Hand X-ray**	
	ACO	**SACO**	**ACO**	**SACO**	**ACO**	**SACO**
			1200 iterations			
Sin	1.5863	1.5729	0.7123	0.9187	1.0512	0.9153
KH	0.6871	0.8029	0.6905	0.9371	0.1637	0.1637
Chi	0.8281	0.6335	0.5714	0.6184	−0.0476	0.0799
			30,000 iterations			
Sin	1.5981	1.5746	0.8801	0.9690	0.9187	0.9153
KH	0.8096	0.8046	0.8499	0.9354	0.1553	0.1755
Chi	0.5949	0.6351	0.5596	0.6200	0.0262	0.0799
			300,000 iterations			
Sin	1.5528	1.5746	0.7777	0.9656	0.9338	0.9153
KH	0.7928	0.8046	0.7777	0.9371	0.1855	0.1755
Chi	0.6569	0.6335	0.5781	0.6200	−0.0443	0.0799

The best solutions obtained for the considered medical images (Head CT, Brain CT and Hand X-ray) while comparing ACO and SACO for 300,000 iterations and the considered demicontractive operators are included in Figure 2; in the last image, the original medical images are overlapped with the best solutions.

Analysis. The values are already standardized based on the denoised ACO and SACO, Canny, Prewitt, Sobel, and Roberts results; therefore, the difference between SACO and ACO is significant from an analytic perspective.

- *An operator comparison analysis* based on Figure 3 uses the difference of SACO vs. ACO values; for the Sin-operator, the difference has a majority of negative values, with ACO obtaining better values than SACO for the considered operators (44.45%); for the other two operators, χ-operator (Chi) and KH-operator, 77.78% shows SACO performing better than ACO on the 9 considered cases of medical images;
- *Medical image analysis.* For each medical image, including head CTs, brain CTs and hand X-rays, Figure 3 identifies operators' behavior on the difference between SACO and AC0. The lowest SACO performance, 44.44%, was obtained for the head CT medical images; its highest performance was 100% for the brain CT images, while for the hand X-rays, a 55.56% performance value was obtained. The percentage is based on the number of considered medical images.

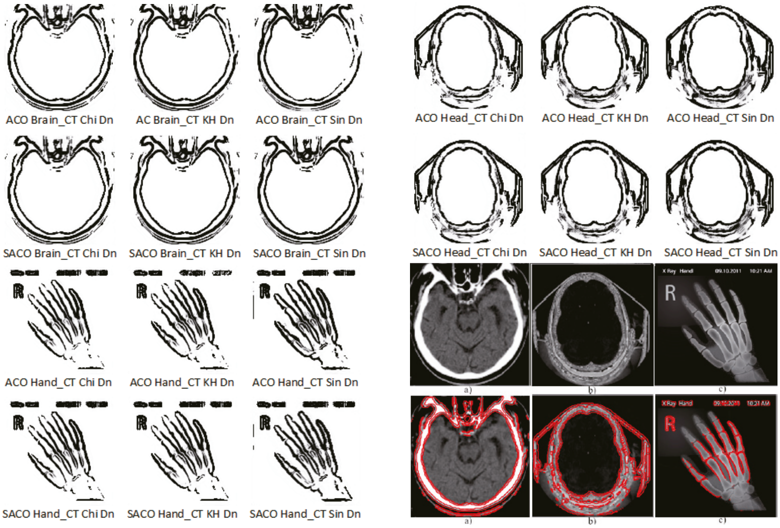


Figure 2. Successive illustrations of the best solutions obtained after 300,000 iterations with *Ant colony optimization (ACO)* and *Sensitive ACO (SACO)* post-processed with the *Denoise Convolutional Neural Network (DnCNN)* and the overlapped best solutions' edges over the original medical images for (**a**) Brain CT; (**b**) Head CT and (**c**) Hand X-ray.

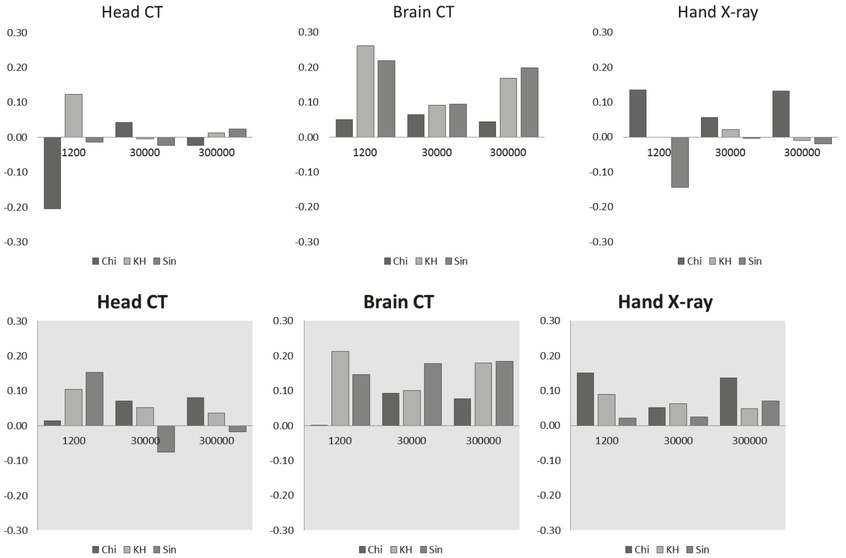


Figure 3. Head CT, Brain CT and Hand X-ray results based on the difference between SACO and ACO standardized values, before (up) and after post-processing with the DeNoise convolutional neural network (DnCNN) (down).

Stability & quality of the solutions. The quality of the partial solution is influenced by the amount of modified pheromone of the ants' trail.

The stability of the global solution is influenced by used parameters. The included PSL parameter hopefully influenced the global solution for the better.

The global solution is found after the entire ant colony, based on the existing pheromone information, is guided to more promising regions in the search space.

The pheromone sensitivity factor balances the exploring and exploiting activities; its value is a number from $[0, 1]$. An ant ignores information when $PSL = 0$ and has the maximum pheromone sensitivity when $PSL = 1$.

- *An exploring search* is made by independent ants with a low PSL value.
- *An exploiting search* is made by sensitive ants to pheromone traces, the intensively exploitative ants with a high PSL value.

In time, the process modifies ants' pheromone sensitivity (PSL). In the current work, the PSL is globally modified (increased or decreased) by the search space topology [17].

SACO advantages. By adding the PSL vector, the present algorithm offers the stability of its solutions; for the considered examples, after around 300 iterations, SACO generated edges which almost overlapped over the original images. As a plus, the image edge results are much more compact and close to the original when compared with the ACO results.

SACO disadvantages. As the number of parameters increases, the user should properly configure their values. This could take more time and resources, but the improved results are worth the effort.

Figure 4 shows the improvements of SACO-DnCNN compared to the Canny, Prewitt, Sobel and Roberts edge detection techniques [32]. The best SACO-DnCNN results, using χ, outperform the Canny Edge detector results by 37.76%; the Prewitt, Sobel and Roberts [32] methods were significantly outperformed by over 159%, 157% and 224%.

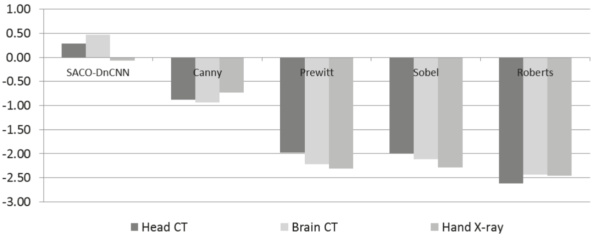


Figure 4. Comparison of the best SACO-DnCNN, Canny, Prewitt, Sobel, and Roberts methods on the Head CT, Brain CT and Hand X-ray medical images.

Future work will use images with higher resolutions, and hopefully the impact of sensitivity will improve the problem solutions.

Future work will also include implementing specific ACO and SACO features to solve publicly available medical datasets, including COVID- and SARS-Cov-2-related data sets. Furthermore, sensitivity for different artificial intelligence models could be involved within different domains, e.g., data mining [33] and similar.

Other improvements could utilize fuzzy techniques and multiple ant colonies for ACO, as in [34], which could be used to enhance the solutions for image edge detection.

Further work could make use of image segmentation with edge detection in order to obtain a more thorough edge [35]. As prerequisites for the development of knowledge-based applications, ontologies for the segmentation of radiological images [36] were proposed by the authors.

Other metaheuristics, mostly bio-inspired ones [37], could be further enhanced with sensitivity features in order to improve the results of complex problems. Human-in-the-loop [38] could also enhance the problem results.

5. Conclusions

Medical image edge detection is nowadays a must in the context of pandemics and other illness and injuries. As classical algorithms have less performance within image edge detection, metaheuristics are used for feasible solutions. The current paper shows the efficiency of bio-inspired algorithms, in particular of the ant-based technique, with an emphasis on the sensitive version of *Ant Colony Optimization (ACO)*.

Sensitivity plays a crucial role in the exploration and exploitation of ants' solutions within the environment; the sensitivity level starts with the maximum level of sensitivity, one, per the ACO level of sensitivity, and during the processes, the ant's level of sensitivity changes. they become less or more sensitive to the environment based on the PSL probability, which is influenced by ants behavior and image intensity.

Nevertheless, the demicontractive operators shows their utility in edge detection problems; an analysis of the results with the presented operators shows how the results vary based on the operators' features.

The edges obtained with each considered operator were overlapped over the original images. The majority of edges were superposed, following the CT and X-ray original bone lines.

Author Contributions: Conceptualization, investigation C.T. and C.-M.P.; methodology formal analysis C.T. and O.M., software, C.T.; validation C.T. and O.M.; writing—original draft preparation C.T. and C.-M.P. and O.M.; writing—review and editing C.T. and C.-M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant Colony Optimization
SACO	Sensitive Ant Colony Optimization
DnCNN	Denoise Convolutional Neural Network

References

1. Dorigo, M.; Stützle, T. *Ant Colony Optimization*; MIT Press: Cambridge, MA, USA, 2004.
2. Marginean, A.N.; Muntean, D.D.; Muntean, G.A.; Priscu, A.; Groza, A.; Slavescu, R.R.; Pintea, C.M. Reliable learning with PDE-based CNNs and dense nets for detecting COVID-19, pneumonia, and tuberculosis from chest X-ray images. *Mathematics* **2021**, *9*, 434. [CrossRef]
3. Chattopadhyay, S.; Dey, A.; Singh, P.K.; Geem, Z.W.; Sarkar, R. Covid-19 Detection by Optimizing Deep Residual Features with Improved Clustering-Based Golden Ratio Optimizer. *Diagnostics* **2021**, *11*, 315. [CrossRef] [PubMed]
4. Castiglione, A.; Vijayakumar, P.; Nappi, M.; Sadiq, S.; Umer, M. COVID-19: Automatic Detection of the Novel Coronavirus Disease From CT Images Using an Optimized Convolutional Neural Network. *IEEE Trans. Ind. Inform.* **2021**, *17*, 6480–6488. [CrossRef]
5. Voβ, S.; Martello, S.I.H.; Roucairol, C. (Eds.) *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*; Publisher: Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
6. Voβ, S. Meta-heuristics: The state of the art. In *Workshop on Local Search for Planning and Scheduling*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–23.
7. Liantoni, F.; Rozi, N.F.; Indriyani, T.; Rahmawati, W.M.; Hapsari, R.K. Gradient based ant spread modification on ant colony optimization method for retinal blood vessel edge detection. *Iop Conf. Ser. Mater. Sci. Eng.* **2021**, *1010*, 012021. [CrossRef]
8. Li, J.; An, X. Efficient Filtering for Edge Extraction under Perspective Effect. *Appl. Sci.* **2021**, *11*, 8558. [CrossRef]
9. Crisan, G.C.; Nechita, E.; Palade, V. Ant-based system analysis on the traveling salesman problem under real-world settings. In *Combinations of Intelligent Methods and Applications*; Springer: Cham, Switzerland, 2016; pp. 39–59.
10. Paprocka, I.; Krenczyk, D.; Burduk, A. The Method of Production Scheduling with Uncertainties Using the Ants Colony Optimisation. *Appl. Sci.* **2021**, *11*, 171. [CrossRef]
11. Matei, O.; Rudolf, E.; Pintea, C.M. Selective Survey: Most Efficient Models and Solvers for Integrative Multimodal Transport. *Informatica* **2021**, *32*, 371–396. [CrossRef]
12. Vescan, A.; Pintea, C.M.; Pop, P.C. Test Case Prioritization-ANT Algorithm with Faults Severity. *Logic J. IGPL* **2020**, *29*, jzaa061. [CrossRef]
13. Pintea, C.-M.; Pop, P.C. Sensor networks security based on sensitive robots agents. A conceptual model. *Adv. Intell. Syst. Comput.* **2013** *189*, 47–56. [CrossRef]

14. Pintea, C.-M.; Pop, P.C. Sensitive Ants for Denial Jamming Attack on Wireless Sensor Network. *Adv. Intell. Soft Comput.* **2014**, *239*, 409–418. [CrossRef]
15. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]
16. Zhang, Z.; Liu, Y.; Liu, T.; Li, Y.; Ye, W. Edge Detection Algorithm of a Symmetric Difference Kernel SAR. Image Based on the GAN Network Model. *Symmetry* **2019**, *11*, 557. [CrossRef]
17. Pintea, C.-M.; Ticala, C. Medical image processing: A brief survey and a new theoretical hybrid ACO model. In *Combinations of Intelligent Methods and Applications. Smart Innovation, Systems and Technologies*; Springer: Cham, Switzerland, 2016; Volume 46, pp. 117–134. [CrossRef]
18. Ticala, C.; Zelina, I.; Pintea, C.-M. Admissible Perturbation of Demicontractive Operators within Ant Algorithms for Medical Images Edge Detection. *Mathematics* **2020**, *8*, 1040. [CrossRef]
19. Rus, I.A. An abstract point of view on iterative approximation of fixed points. *Fixed Point Theory* **2012**, *33*, 179–192.
20. Berinde, V.; Ticala, C. Enhancing Ant-Based Algorithms for Medical Image Edge Detection by Admissible Perturbations of Demicontractive Mappings. *Symmetry* **2021**, *13*, 885. [CrossRef]
21. Ticala, C. A weak convergence theorem for a Krasnoselskij type fixed point iterative method in Hilbert spaces using an admissible perturbation. *Sci. Stud. Res.* **2015**, *25*, 243–252.
22. Tian, J.; Yu, W.; Xie, S. An ant colony optimization algorithm for image edge detection. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 751–756.
23. Ticala, C.; Zelina, I. New ant colony optimization algorithm in medical images edge detection. *Creat. Math. Inf.* **2020**, *29*, 101–108. [CrossRef]
24. Pintea, C.M.; Chira, C.; Dumitrescu, D.; Pop, P.C. A sensitive metaheuristic for solving a large optimization problem. *Lect. Notes Comput. Sci.* **2008**, *4910*, 551–559. [CrossRef]
25. Chira, C.; Dumitrescu, D.; Pintea, C.M. Learning sensitive stigmergic agents for solving complex problems. *Comput. Inform.* **2010**, *29*, 337–356.
26. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
27. Kanchi-Tian, J. *Image Edge Detection Using Ant Colony Optimization Version 1.2.0.0*; MATLAB Central File Exchange; University of Science & Technology: Wuhan, China, 2011.
28. Edge Function. MATLAB Central File Exchange. Available online: https://www.mathworks.com/help/images/ref/edge.html (accessed on 5 August 2021).
29. X-ray Hand. Vista Medical Pack. License: Free for Non Commercial Use. p. 236487. Available online: https://www.iconspedia.com/ (accessed on 5 August 2021).
30. Head CT. Online Medical Free Image. Available online: http://www.libpng.org/pub/png/pngvrml/ct2.9-128x128.png (accessed on 5 August 2021).
31. Denoise Image Using Deep Neural Network. MATLAB Central File Exchange. Available online: https://www.mathworks.com/help/images/ref/denoiseimage.html (accessed on 5 August 2021).
32. Kumar, S.; Upadhyay, A.K.; Dubey, P.; Varshney, S. Comparative analysis for Edge Detection Techniques. In Proceedings of the 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 19–20 February 2021; pp. 675–681.
33. Avram, A.; Matei, O.; Pintea, C.; Anton, C. Innovative Platform for Designing Hybrid Collaborative & Context-Aware Data Mining Scenarios. *Mathematics* **2020**, *8*, 684. [CrossRef]
34. Pintea, C.M.; Matei, O.; Ramadan, R.A.; Pavone, M.; Niazi, M.; Azar, A.T. A Fuzzy Approach of Sensitivity for Multiple Colonies on Ant Colony Optimization. *Soft Comput. Appl.* **2016**, *634*, 87–95. [CrossRef]
35. Ahn, E.; Kim, J.; Bi, L.; Kumar, A.; Li, C.; Fulham, M.; Feng, D.D. Saliency-Based Lesion Segmentation via Background Detection in Dermoscopic Images. *IEEE J. Biomed. Health Inform.* **2017**, *21*, 1685–1693. [CrossRef] [PubMed]
36. Matei, O. Defining an ontology for the radiograph images segmentation. In Proceedings of the 9th International Conference on Development and Application Systems, Suceava, Romania, 22–24 May 2008; pp. 266–271.
37. Abd, E.M.; Ewees, A.A.; Ibrahim, R.A.; Lu, S. Opposition-based moth-flame optimization improved by differential evolution for feature selection. *Math. Comput. Simul.* **2020**, *168*, 48–75.
38. Holzinger, A.; Plass, M.; Kickmeier-Rust, M.; Holzinger, K.; Crişan, G.C.; Pintea, C.M.; Palade, V. Interactive machine learning: Experimental evidence for the human in the algorithmic loop: A case study on Ant Colony Optimization. *Appl. Intell.* **2019**, *49*, 2401–2414. [CrossRef]

symmetry

Article

An Image Recognition Method for Coal Gangue Based on ASGS-CWOA and BP Neural Network

Dongxing Wang [1,2,3], Jingxiu Ni [4,*] and Tingyu Du [3,*]

[1] R & D Department, Zhuhai Xinhe Technology Co., Ltd., Zhuhai 519600, China; wangdongxing85@163.com or 0621169@zju.edu.cn
[2] School of Electrical Engineering, Zhejiang University, Hangzhou 310000, China
[3] School of Mechanical Electronic & Information Engineering, China University of Mining & Technology-Beijing, Beijing 100083, China
[4] Comprehensive Experimental Teaching Demonstration Center of Engineering, Beijing Union University, Beijing 100101, China
* Correspondence: njx1211@163.com or jingxiu@buu.edu.cn (J.N.); dutingyu666@163.com or bqt1900405016@cumtb.edu.cn (T.D.)

Abstract: To improve the recognition accuracy of coal gangue images with the back propagation (BP) neural network, a coal gangue image recognition method based on BP neural network and ASGS-CWOA (ASGS-CWOA-BP) was proposed, which makes two key contributions. Firstly, a new feature extraction method for the unique features of coal and gangue images is proposed, known as "*Encircle–City Feature*". Additionally, a method that applied ASGS-CWOA to optimize the parameters of the BP neural network was introduced to address to the issue of its low accuracy in coal gangue image recognition, and a BP neural network with a simple structure and reduced computational consumption was designed. The experimental results showed that the proposed method outperformed the other six comparison methods, with recognition of 95.47% and 94.37% in the training set and the test set, respectively, showing good symmetry.

Keywords: coal gangue image; classification; wolf pack optimization; BP neural network

Citation: Wang, D.; Ni, J.; Du, T. An Image Recognition Method for Coal Gangue Based on ASGS-CWOA and BP Neural Network. *Symmetry* **2022**, *14*, 880. https://doi.org/10.3390/sym14050880

Academic Editor: Mihai Postolache

Received: 24 March 2022
Accepted: 22 April 2022
Published: 25 April 2022

1. Introduction

Removing gangue from raw coal is conducive to improving the efficiency of coal utilization and reducing environmental pollution. Therefore, the identification of gangue and coal blocks is a necessary step for the efficient removal of coal gangue in coal production, which is of great significance for environmental protection [1,2]. Because of its obvious advantages of resource saving, low cost, the simple processing system and convenient maintenance, image recognition technology for the separation of coal gangue has been widely studied and applied in the literature [3–7]. Researchers have developed a variety of image recognition methods for coal gangue image recognition, which contribute to research into coal gangue separation technology based on image recognition and have laid a foundation for further research. However, these methods have their shortcomings. For example, in [3], the proposed method resulted in a waste of computing resources and low time efficiency because it calculated 15 eigenvalues at the same time, although only the best five eigenvalues were actually used. In the study by Zhou et al. [4], the new method adopted a deep convolution neural network to address the task of online accurate and rapid identification of coal gangue, which required extensive computation. In other studies [5,6], although the recognition rate of the methods used was high, the artificially set threshold parameters played a decisive role in the intermediate calculation link, which depended on an extensive experimental statistical analysis or experience, resulting in the calculation being complex. The stability of performance was questionable, and the generalized ability to recognize coal gangue images with different characteristics in different regions was not necessarily high. The authors of [7], proposed a method that required a large number of samples generated by the introduction of the transfer learning method, suggesting that the method's generalization and universality are questionable and may be limited when the sample size is small.

Since it was proposed by the team of Rumelhart and McClelland in 1986, the BP (back propagation) neural network has been widely used in the field of image recognition [8] because of its simple structure and high calculation efficiency, such as in [9]. To solve the problem of remote sensing image classification, the authors established an improved BP neural network and set the dynamic training intensity to improve the learning speed and classification accuracy of the BP neural network classifier. However, determining the optimal training intensity consumes a large amount of computing resources; moreover, the sample set classification problems of different fields have different optimal training intensities due to the large differences in the sample characteristics, so the generalizability is questionable, probably resulting in this method being unsuitable for image coal gangue classification with a wide area and many sources. In another study [10], a small image recognition and classification method based on GA-BP was proposed, in which the combination of a genetic algorithm and the BP neural network gave full play to the nonlinear mapping ability of the neural network, resulting in strong learning ability and fast convergence speed. The experimental results showed that the new method had higher recognition accuracy and better performance than the traditional BP algorithm and the GA-BP algorithm. The authors of [11] adopted a hybrid algorithm combining a genetic algorithm and a back propagation algorithm, and the experimental results showed that this GA-BP algorithm had higher efficiency, robustness and practicability. The researchers in [12] proposed an epilepsy diagnosis method based on an improved genetic algorithm–optimized back propagation (IGA-BP) neural network and used this method to detect clinical epilepsy quickly and effectively. Liu et al. [13] used a genetic algorithm (GA) to construct a classification model based on the BP neural network to automatically identify the correlations in a multi-mode blog, and the experimental results showed that the classification model based on GA-BP was better than the traditional BP neural network. Yu et al. [14] considered the problem that a BP algorithm based on a gradient descent principle falls into the local minima, and thus, the classification of multispectral remote sensing images using spectral information cannot obtain ideal results. Yu et al. developed a new method combining feature texture knowledge with a BP neural network trained by particle swarm optimization (PSO). The experimental results showed that this method had improved classification accuracy. Ying et al. [15] proposed a method combining a random loss algorithm and particle swarm optimization (PSO-BP) for the recognition and classification of small images, which corrected the weights of the PSO algorithm based on the error back propagation adjustment of the traditional BP algorithm to establish the PSO-BP network model. The hidden layer unit of the PSO-BP network was improved by using the random loss algorithm, thus achieving faster operation speeds. The authors of [16,17] proposed variants of the PSO-BP combining a BP neural network and PSO to address the task of supervised classification of synthetic aperture radar (SAR) images and the problems of low accuracy and efficiency in traditional part classification methods. This kind of BP neural network method based on GA or PSO optimization has a certain reference value for coal gangue image classification. Unfortunately, the accuracy of image coal gangue classification was still not high, as shown by a comparative experiment in a follow-up paper. In addition, the BP neural network has also been applied in other fields. For example, in [18], the researchers proposed a metal surface defect classification method based on an improved bat algorithm to optimize the BP neural network, which was used to classify images of defects with different characteristics. The researchers in [19] proposed an automatic recognition method of cloud and precipitation particle shapes based on a BP neural network to solve the problem that the shape of cloud particle images measured by airborne cloud imaging probes (CIPs) cannot be automatically recognized.

Although these methods have some shortcomings, they have still contributed to the research into coal gangue separation technology based on image recognition and laid a foundation for further research. In 2007, Yang and his coauthors proposed a new swarm intelligence algorithm [20], which simulates the predation process of wolves to solve complex nonlinear optimization problems. Due to its superior performance, it has been widely used in various fields and has been continuously developed and improved. For example, in [21], the author proposed a novel and effective opposite wolf pack algorithm to estimate the parameters of a Lorenz chaotic system. In another study [22], the improved wolf pack search algorithm was used to calculate the quasi-optimal trajectory of a rotor UAV in complex three-dimensional space. Moreover, in [23], the wolf pack algorithm was used to find the root of the polynomial equation of the problem accurately and quickly. Similarly, in [24], Zhou Qiang and others proposed a wolf pack search algorithm based on the leader strategy (LWCA). The researchers in [25] designed an adaptive shrinking grid search chaos wolf optimization algorithm (ASGS-CWOA) using adaptive standard deviation updating to achieve better performance, which included new regeneration rules, a new raid strategy, a new siege strategy and a new siege adaptive step size. Inspired by the good optimization ability of the wolf pack intelligent optimization algorithm and the excellent classification performance of the BP neural network, this article proposes a coal gangue image recognition method based on ASGS-CWOA and the BP neural network.

The remainder of this article is organized as follows. Section 2 presents a description of "*Encircle–City Feature*", a combination of ASGS-CWOA and the BP neural network and an overview of new method. In Section 3, we present comparative experiments to show the effectiveness of the proposed approach. The conclusion is given in Section 4.

2. Proposed Method

2.1. Encircle–City Feature

By comparing a large number of sample images, it was found that the brightness area of coal block images is significantly larger than that of gangue images, but the degree of recognition of distinguishing coal blocks and gangue according to the contrast index is not very high because the contrast of coal is significantly greater than that of gangue from a local point of view. Nevertheless, this difference in contrast will partially offset the images from each other from an overall point of view.

In this article, the basic idea of *Encircle–City Feature* is to divide the sample image into several continuous small areas of 50 × 50 without overlaps or blanks, in which the following operations should be performed. If we assume that the matrix of the small 50 × 50 area is I, the implementation steps are as follows:

Step 1: Divide each sample image evenly into $M \times N$ small areas with M rows and N columns such that each small area should be 50 pixels × 50 pixels without overlaps and blanks and perform Steps 2 to 4 for each one.

Step 2: Obtain the average gray value of the "*City*". The total gray (denoted by $City_{sum\_gray}$) of small 30 × 30 areas is calculated to obtain the average gray (denoted by $City_{average\_gray}$) in the central district. This is like a castle located in the central area, so we call it the "City" (shown in red in Figure 1). This is given by Equation (1).

$$\begin{cases} City_{sum\_gray} = \sum_{i=11,j=11}^{40,40} I(i,j) \\ City_{average\_gray} = {}^{City_{sum\_gray}}/900 \end{cases} \tag{1}$$

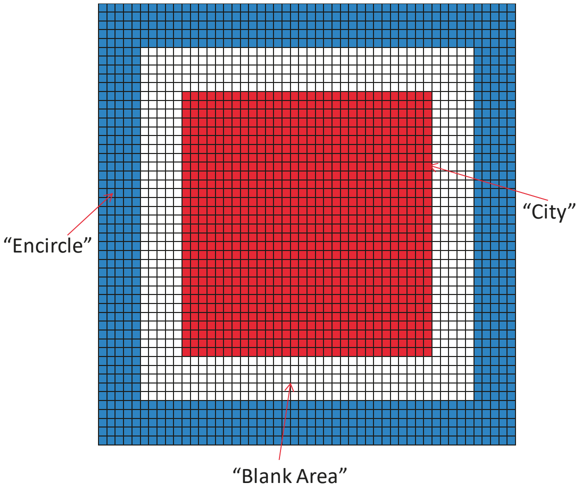


Figure 1. Schematic diagram of *Encircle—City Feature*.

Step 3: Obtain the total gray value (denoted by $Encircle_{sum\text{-}gray}$) and the average gray value (denoted by $Encircle_{average\text{-}gray}$). The peripheral part of the small 50 × 50 area excluding the central 40 × 40 pixels is like the wall around a castle, so it is called the "*Encircle*", as shown in blue in Figure 1. Equation (2) is as follows:

$$\begin{cases} Encircle_{sum-gray} = \sum_{i=1,j=1}^{50,50} I(i,j) - \sum_{i=6,j=6}^{45,45} I(i,j) \\ Encircle_{average-gray} = Encircle_{sum\_gray}/900 \end{cases} \tag{2}$$

Step 4: For the small area of Row m and Column n, obtain the "*Encircle–City*" value using Equation (3):

$$\begin{cases} Encircle - City(m,n) = City_{average-gray} - Encircle_{average-gray} \\ (m <= M, n <= N) \end{cases} \tag{3}$$

Step 5: Obtain the average value of the "*Encircle–City*" matrix ($M \times N$ matrix "*Encircle–City*" obtained using Step 4). Finally, calculate the "*Encircle–City*" value of the whole sample image as shown in Equation (4).

$$\begin{cases} Average_{gray} = \frac{\sum_{m=1,n=1}^{M,N} Encircle-City(m,n)}{(M \times N)} \\ Length = find - length(Encircle - City > Average_{gray}) \\ Encircle - City_{eigenvalue} = length/(M \times N) \end{cases} \tag{4}$$

where "$Average_{gray}$" means the average value of the overall matrix "*Encircle–City*"; "*Length*" means the number of elements larger than "$Average_{gray}$" in the matrix "*Encircle–City*", which is obtained by the function "*find-length*"; $Encircle\text{–}City_{eigenvalue}$ stands for the overall "*Encircle–City Feature*" value of one sample image, as shown in Figure 2.

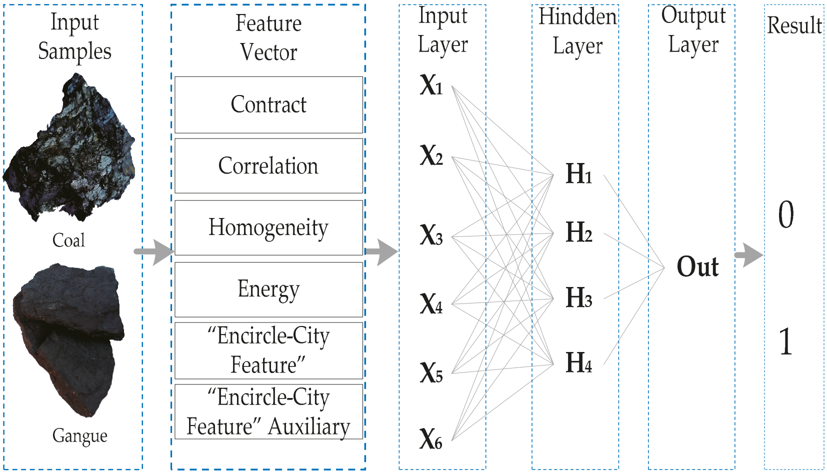


Figure 2. Recognition network for coal and gangue images.

In this article, small areas of 50 × 50 were used to segment one overall sample image, in which "*Encircle*" contains 900 pixels and "*City*" contains 900 pixels, meaning that the *Encircle–City Feature* can better reflect the texture features of the image under ideal circumstances. It should be noted that we only used the *Encircle–City Feature* value to identify coal and gangue, and the recognition accuracy reached 83.24%, which is not discussed in detail due to limited space.

Unfortunately, the images are often irregular such that the "*Encircle*" and the "*City*" in the small 50 × 50 local area do not necessarily strictly follow the ideal situation shown in Figure 2, which leads to the lower accuracy in identifying coal and coal gangue images using the *Encircle–City Feature* (83.24%). In fact, we are more concerned about the light–dark contrast of small local areas than the whole image, so we introduced the auxiliary value of the "*Encircle–City Feature*": the "*Encircle–City Assist*", the details of which are given below.

Step 1: Divide each sample image evenly into small $M \times N$ areas with M rows and N columns such that each small area must include 50 pixels × 50 pixels without overlaps or blanks and perform Steps 2 to 4 for each one.

Step 2: Sort the image pixels of the small 50 × 50 area in ascending order according to the gray value, as shown in Equation (5):

$$block_{sort} = Sort(block) \tag{5}$$

where "$block_{sort}$" means the matrix after arrangement in ascending, which is calculated and returned by the function "*Sort*".

Step 3: Calculate "*Encircle–City*$_{Assist}$", the auxiliary value of the "*Encircle–City Feature*" of the current small 50 × 50 area block, which is the difference between the second half of "$block_{sort}$" and its first half, as shown in Equation (6).

$$Encircle - City_{assist} = block(2501:5000) - block(1:2500) \tag{6}$$

It should be noted that we only used the "*Encircle–City*$_{Assist}$" value to identify coal and gangue, and the recognition accuracy reached 78.21%, which is not discussed in detail due to limited space.

2.2. ASGS-CWOA-BP

2.2.1. Overview of ASGS-CWOA

We proposed ASGS-CWOA in [20] with three contributions: the strategy of adaptive shrinking grid search (ASGS), the strategy of opposite–middle raid (OMR) and the adaptive standard deviation updating amount (ASDUA), which has been shown to have superior performance compared with some state-of-the-art algorithms at that time. Accordingly, in this study, we use ASGS-CWOA to address the issue of optimizing the weights of the BP neural network for coal gangue image recognition. In order to adapt to the particularity of the recognition network weights, which are always small, some necessary adjustments of the step size should be made according to the following rules.

In this article, the variation range of the weights was set between −5 and 5 based on experience, i.e., *range_max* = 5 and *range_min* = −5. Correspondingly, the value range is [−5, 5] in any dimension for the position of one wolf.

Thus, the step size of the siege stage can be obtained by using Equation (7) as follows:

$$\begin{cases} ostep\_c\_max = (range\_max - range\_min)/2 \\ step\_c\_min = 0.01 \\ stepc = step\_c\_min \times (range\_max - range\_min) \times exp((log(step\_c\_min/step\_c\_max)) \times t/T) \end{cases} \tag{7}$$

where *step_c_max* is the upper limit of the siege step's size, *step_c_min* is the lower limit, *t* indicates the current number of iterations and *T* represents the upper limit.

The step sizes of the migration stage and the summons–raid stage can be obtained by using Equation (8) as follows:

$$\begin{cases} stepa = stepc \times 100,\ when\ stepc\ \geq\ 0.001 \\ stepa = stepc \times 1000,\ when\ stepc < 0.001 \\ stepb = stepa \times 2 \end{cases} \tag{8}$$

where *stepa* means the step size of the migration stage and *stepc* means one of the summons–raid stages. In order to prevent the *stepc* value from getting smaller and smaller with each iteration such that the values of *stepa* and *stepb* become too small to affect the optimization effect, the values of *stepa* and *stepb* are amplified when the value of *stepc* is less than 0.001.

2.2.2. The Recognition Network

Based on the BP neural network and the ASGS-CWOA algorithm and considering the factors of low network complexity and less computation, this research designed a recognition network with a simple structure for coal and gangue images (RN-CGI), which includes six input layers, four hidden layers and one output layer, as shown in Figure 2.

Here, the hidden layer adopts the "tansig" kernel function, and the output layer adopts the "purelin" kernel function. The position coordinates of each wolf in the wolf pack represent the weights of the BP neural network, and the fitness value is jointly calculated by the recognition network and the sample eigenvector according to Equations (9) and (10).

$$X_i = (x_{i1}, \ldots, x_{id}, \ldots, x_{iD})\ (i = 1, \ldots, n;\ d = 1, \ldots, D) \tag{9}$$

$$\begin{cases} Net.W = \begin{bmatrix} x_{i1} & x_{i5} & x_{i9} & x_{i13} & x_{i17} & x_{i21} \\ x_{i2} & x_{i6} & x_{i10} & x_{i14} & x_{i18} & x_{i22} \\ x_{i3} & x_{i7} & x_{i11} & x_{i15} & x_{i19} & x_{i23} \\ x_{i4} & x_{i8} & x_{i12} & x_{i16} & x_{i20} & x_{i24} \end{bmatrix} \\ Net.L = \begin{bmatrix} x_{i25} \\ x_{i26} \\ x_{i27} \\ x_{i28} \end{bmatrix} \end{cases} \tag{10}$$

where X_{id} is the coordinate of the i-th wolf in the d-th dimension, *Net.W* is the weight from the input layer to the hidden layer, and *Net.L* is the weight from the hidden layer to the output layer. From the sum of the elements of *Net.W* and *Net.L*, it is obvious that D is 28, correspondingly. In this way, the location information of each wolf can be mapped into the weight parameters of the recognition network. By continuously optimizing the location information of the wolves, the potential optimal solution with the best fitness value is obtained.

In this article, the network output values of all training samples are calculated according to the network weight parameters mapped by the position information of $wolf_i$. Since this article considers the binary classification of coal and gangue images (coal = 0; gangue = 1) and the BP neural network adopts the "tansig" and "purelin" kernel functions, the following judgment can be made for the network output value out_i: for the i-th sample, when out_i is less than 0.5, this indicates coal, i.e., set 0, but if it is greater than or equal to 0.5, it is judged to be gangue, i.e., set 1. Accordingly, the fitness function can be given by Equation (11).

$$\left\{\begin{array}{c} out = (out_1, out_2, \ldots\ldots, out_i, \ldots\ldots out_{num}),\ i = 1, 2, \ldots, num \\ right\_num = length(out = BJ) \\ fitness_k = right\_num/num \end{array}\right. \tag{11}$$

where *num* is the number of training or test samples, *right_num* is the number of samples correctly identified, *BJ* is the label (0 or 1) of the training or test sample, length (*out* = *BJ*) is a function that can calculate and return the number of correctly identified samples and $fitness_k$ is the fitness value of $wolf_k$ of the k-th wolf, that is, the recognition accuracy.

2.3. Overview of the Proposed Method

2.3.1. Image Preprocessing

With a camera (Huawei Honor 20, 48 million pixels), 358 gangue images and coal block images (including 285 gangue images and 173 coal block images) were taken. The image size was 4000 × 3000, forming the original sample set. We preprocessed the original images in order to extract the feature vectors, as shown in Figure 3.

1. Image Graying: Grayscale images refer to images containing only brightness information and no color information. Grayscale processing is the process of changing the color image containing brightness and color into grayscale images.
2. Median Filtering: Median filtering is a nonlinear signal processing technology that can effectively suppress noise based on the sorting statistical theory. Its basic principle is to replace the gray value of a point in the digital image with the median value of each point in the local neighborhood of the point. This paper used a 3 × 3 local neighborhood.
3. Otsu Segmentation: The Otsu algorithm is an efficient algorithm for image binarization proposed by the Japanese scholar Otsu in 1979. The principle is to divide the original image into foreground and background images by a threshold. For the foreground, N1, csum and M1 are used to represent the number of points, the quality moment and the average gray level of the foreground under the current threshold, respectively. For the background, N2, sum csum and M2 are used to represent the number of points, the quality moment and the average gray level of the background under the current threshold, respectively. When the optimal threshold is selected, the difference between the background and the foreground should be the greatest.
4. Erosion and Dilation: Erosion is the use of algorithms to corrode the edges of the image. The function is to start off the "burr" on the edge of the target. Inflation uses the algorithm to expand the edges of the image. The function is to fill the edges or internal pits of the target. Having the same amount of erosion and dilation can make the target surface smoother, which is a symmetrical process.

5. Target Area Focusing: The size of the original image of the sample was 4000 × 3000. The processing capacity of image storage and calculation is large, and the blank area accounts for a large proportion, resulting in unnecessary gangue in the resources. In order to lock the effective area, we used the corroded and expanded images to minimize the boundary of the target area, remove the unnecessary background and focus on the foreground target area of the image.
6. Nearest Interpolation Image Size Scaling: After the target area focusing operation, due to the differences in the influence of sample image noise and the different sizes of the target areas, the sizes of the gray image of the "target area" were different. In order to unify the sample size, the size scaling operation was carried out on the image; that is, the "nearest interpolation" operation was used to reduce the size of the sample images that were greater than 800 × 600 and increase the size of the sample images that were less than 800 × 600. The unified sample image size was 800 × 600.

The results of the prepossessing process are shown in Figure 3.

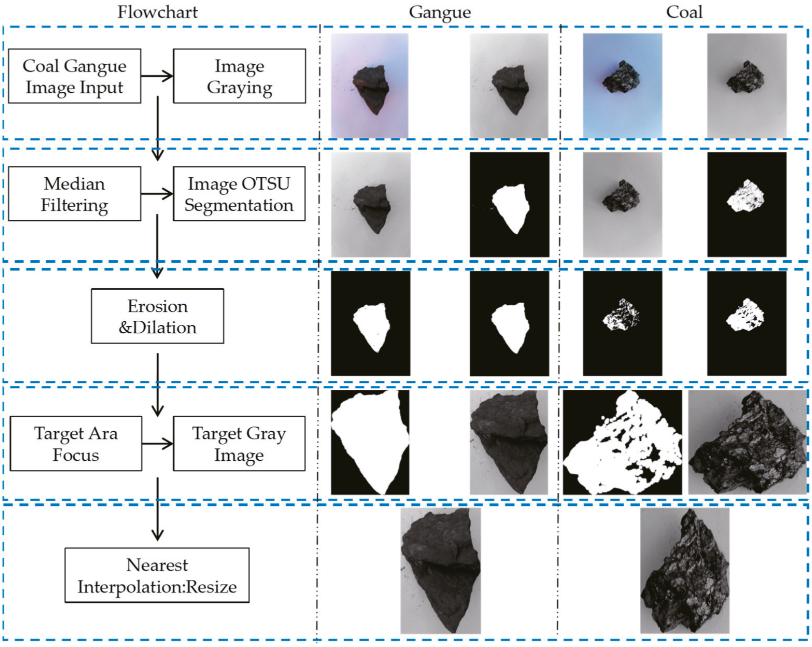


Figure 3. Flowchart of image preprocessing.

2.3.2. Gray Level Co-Occurrence Matrix (GLCM)

The method commonly used to describe the grayscale texture is the grayscale correlation matrix. The index eigenvalues derived from the gray level co-occurrence matrix are as follows: "*contrast*" returns the contrast between a pixel in the whole image and its neighbors. The contrast of an image composed of constants is 0. The calculation equation is

$$Contrast = \sum_{i,j} |i-j|^2 p(i,j) \tag{12}$$

"*Correlation*" returns the cross-correlation between a pixel in the whole image and its neighbors. The value range is [−1, 1]. The cross-correlation of images composed of constants is none. The correlation degrees 1 and −1 correspond to complete positive correlation and complete negative correlation, respectively. The calculation equation is

$$Correlation = \sum_{i,j} \frac{(i - \mu * i)(j - \mu * j)p(i,j)}{\sigma_i * \sigma_j} \tag{13}$$

"*Homogeneity*" reflects the tightness of the distribution of elements in the GLCM relative to the diagonal of the GLCM. The value range is [0, 1]. The homogeneity of a diagonal GLCM is 1. The equation is

$$Homogeneity = \sum_{i,j} \frac{p(i,j)}{1 + |i - j|} \tag{14}$$

"*Energy*" returns the sum of squares of all elements in the GLCM. The value range is [0, 1]. The energy of an image composed of constants is 1. The calculation equation is

$$Energy = \sum_{i,j} p(i,j)^2 \tag{15}$$

2.3.3. Feature Extraction of Coal and Gangue Images

According to the theories detailed above, the feature vector of each sample image is composed of six image features (contrast, correlation, homogeneity, energy, *Encircle–City Feature* and *Encircle–City Feature* auxiliary). As shown in Table 1, the sample included 358 sample images composed of 185 gangue images and 173 coal images.

Table 1. Feature vector of the sample set.

Order	Contract	Correlation	Homogeneity	Energy	*Encircle–City Feature*	*Encircle–City Feature Auxiliary*
1	6.08	0.82	0.66	0.14	0.38	0.49
2	5.4	0.8	0.63	0.09	0.36	0.43
3	2.94	0.88	0.75	0.22	0.37	0.42
4	7.92	0.73	0.59	0.08	0.34	0.41
5	8.09	0.77	0.64	0.13	0.34	0.51
6	7.81	0.79	0.65	0.14	0.33	0.46
				……		
353	3.52	0.89	0.73	0.13	0.3	0.34
354	2.54	0.92	0.78	0.13	0.28	0.28
355	3.13	0.9	0.73	0.09	0.34	0.32
356	3.21	0.91	0.73	0.09	0.31	0.36
357	2.81	0.91	0.73	0.11	0.29	0.28
358	3.97	0.87	0.73	0.17	0.31	0.38

As space is limited, only some data have been listed; the complete list of data is given in the link in Appendix A.

2.3.4. Flowchart of the Proposed Method

As shown Figure 4, the flow chart of the proposed method includes the steps of inputting samples, initialization of the ASGS-CWOA, the optimization process and recording the data. The details about inputting the samples are given in Sections 2.3.2 and 2.3.3, and the details of initialization of the ASGS-CWOA, the optimization process and recording the data are described in Sections 2.2.1 and 2.2.2 above.

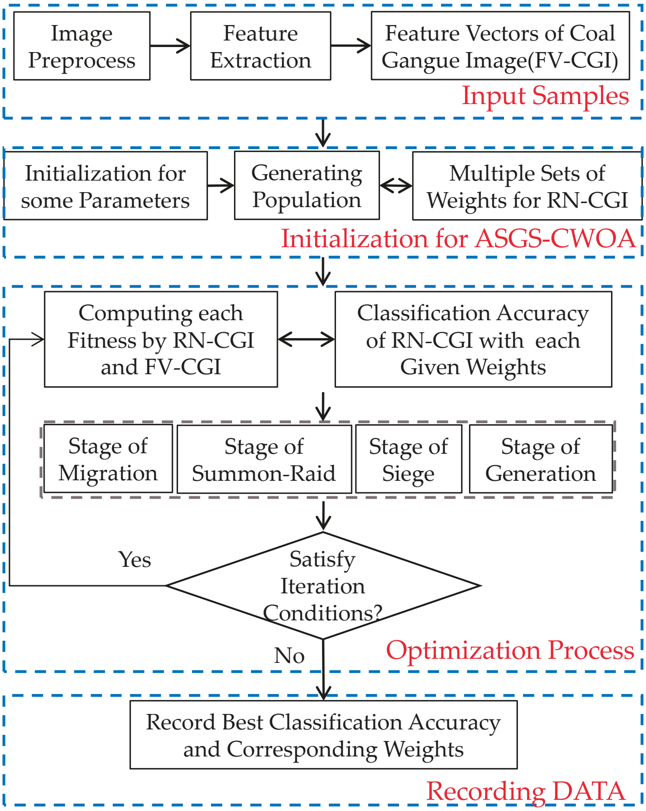


Image Preprocess
Feature Extraction
Feature Vectors of Coal Gangue Image(FV-CGI)
Input Samples
Initialization for some Parameters
Generating Population
Multiple Sets of Weights for RN-CGI
Initialization for ASGS-CWOA
Computing each Fitness by RN-CGI and FV-CGI
Classification Accuracy of RN-CGI with each Given Weights
Stage of Migration
Stage of Summon-Raid
Stage of Siege
Stage of Generation
Yes
Satisfy Iteration Conditions?
No
Optimization Process
Record Best Classification Accuracy and Corresponding Weights
Recording DATA

Figure 4. Flowchart of the proposed method.

3. Simulation Experiment

3.1. Experimental Environment

To verify the feasibility and efficiency of the algorithm proposed in this article, several groups of comparative experiments were carried out by using the new ASGS-CWOA-BP method, the classification method GA-BP based on genetic algorithm optimization and the BP neural network, the classification method PSO-BP based on particle swarm optimization and the BP neural network, the classification method LWCA-BP based on the wolf pack optimization algorithm with the leadership strategy and the BP neural network, and the original BP neural network based on gradient descent and random forest (RF).

Table 1 shows the numerical experimental data based on six-dimensional feature vectors from 358 samples, and Table 2 lists the parameters of the six classification methods for coal and gangue images. The comparative experiments were run on a computer equipped with a CPU (AMD A6-3400m APU with Radeon^{TM} HD Graphics 1.40 GHz), 12.0 GB of memory (11.5 GB available) and Windows 7 (64 bit). To prove the good performance of the proposed algorithm, optimization calculations were run 30 times on the sample feature vectors for testing, and the classification algorithm mentioned above were also tested.

Table 2. Configuration of the other methods in the comparison.

Order	Name	Configuration
1	GP-BP	GA is applied to optimize the BP neural network. The genetic algorithm experiment uses the toolbox of MATLAB 2017A, and its configuration parameters are as follows: the crossover probability is set to 0.7, the mutation probability is set to 0.01 and the generation gap is set to 0.95.
2	PSO-BP	PSO is applied to optimize the parameters of the BP neural network to produce the PSO-BP classification method. The toolbox called "PSOt" in MATLAB is used in experiments of particle swarm optimization, with the following configuration: individual acceleration = 2; weighted initial time = 0.9; weighted convergence time = 0.4. This limits the individual speed to 20% of the variation range.
3	LWCA-BP	LWCA is applied to optimize the parameters of the BP neural network to produce the LWCA-BP method based on the ideas presented in [24]. The configuration is: migration step (Step A) = 1.5, summons–raid step (Step B) = 0.9, siege threshold (R0) = 0.2, upper limit of the siege step (Step c_{max}) = 1×10^6, lower limit of the siege step (Step c_{min}) = 1×10^{-2}, updated amount of the population (M) = 5, maximum number of iterations (T) = 600, number of wolves in the population = 50.
4	BP with Gradient Descent (BP)	BP neural network with gradient descent. This calls the BP neural network training function of MATLAB 2017b to generate the BP network net = newff (P, t, s). The sim (net, in) function is then used to predict the input data. P represents the training sample set, T represents the labels of the training sample set, s represents the network parameters (such as the number of hidden layers), net represents the trained network classification prediction model and in is the input data to be determined.
5	Random Forest (RF)	This calls the random forest function classrf_train of MATLAB 2017b to train the training network and calls classrf_predict to predict the training samples and test samples.
6	ASGS-CWOA-BP	Upper limit number of iterations (T) = 600; number of wolves in the population (N) = 50, Range_max = 5 and Range_min = −5, i.e., the value range is [−5, 5] in any dimension for the position of one wolf.

3.2. Experimental Results

Firstly, Figure 5a,b show the comparison curves for the classification accuracy of each algorithm based on the data in Table 3 for the training set and test set, respectively, from which we can intuitively see that the curve for ASGS-CWOA-BP is better than that of the others, except for the pink curve corresponding to RF on the training set, which means that, on the whole, ASGS-CWOA-BP has the best performance in the classification of coal gangue images using RN-CGI.

It can be clearly seen from Figure 5c that the ASGS-CWOA-BP curve is higher than the GA-BP curve in both the training set and the test set, which means that ASGS-CWOA-BP is better than GA-BP in terms of the accuracy of classifying coal gangue images using RN-CGI. In the same way, ASGS-CWOA-BP is better than PSO-BP and LWCA-BP, as shown in Figure 5d,e, respectively. The principle is that these four methods are based on the same intelligent algorithm to optimize the weight of the BP neural network; however, their classification results are not consistent, which shows that the optimization ability of these four intelligent algorithms is different and that the ability of ASGS-CWOA-BP is the best.

Additionally, Figure 5f indicates that the ASGS-CWOA-BP had better performance than the original BP depending on gradient descent for the classification of coal gangue images using RN-CGI, whether on the training set or the test set. In fact, BP depending on gradient descent had the worst performance compared with GA-BP, PSO-BP, LWCA-BP and ASGS-CWOA-BP, which are based on an intelligent algorithm to optimize the weight of the BP neural network, which shows the inherent deficiency of gradient-descent-based BP resulting from limitations by the degree of the gradient descent because of some particularity of the problem to be solved, while intelligent algorithm-based BPs are not subject to such restrictions.

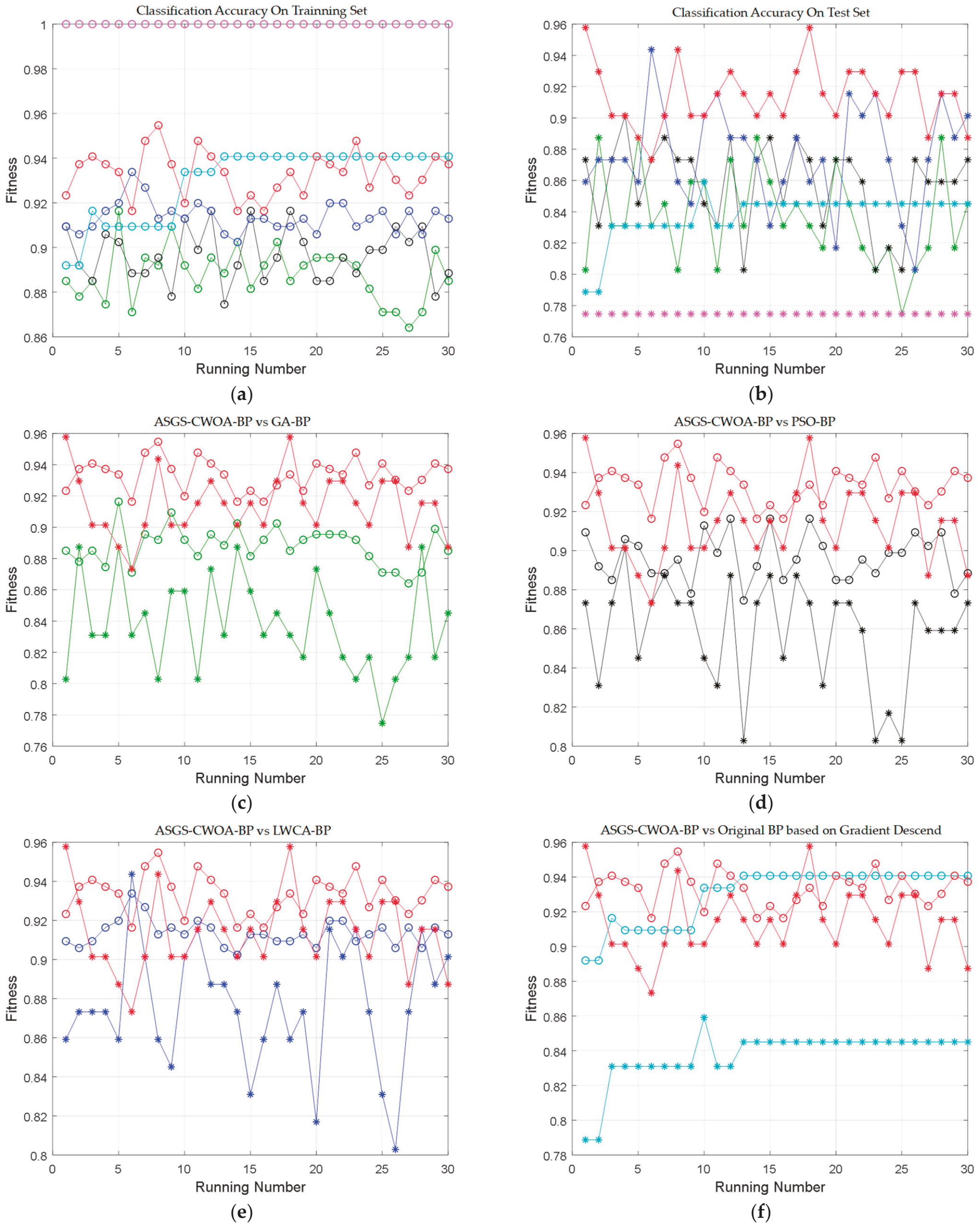
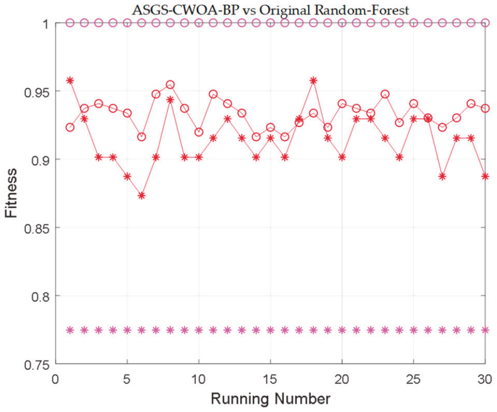


Figure 5. *Cont.*



(**g**)

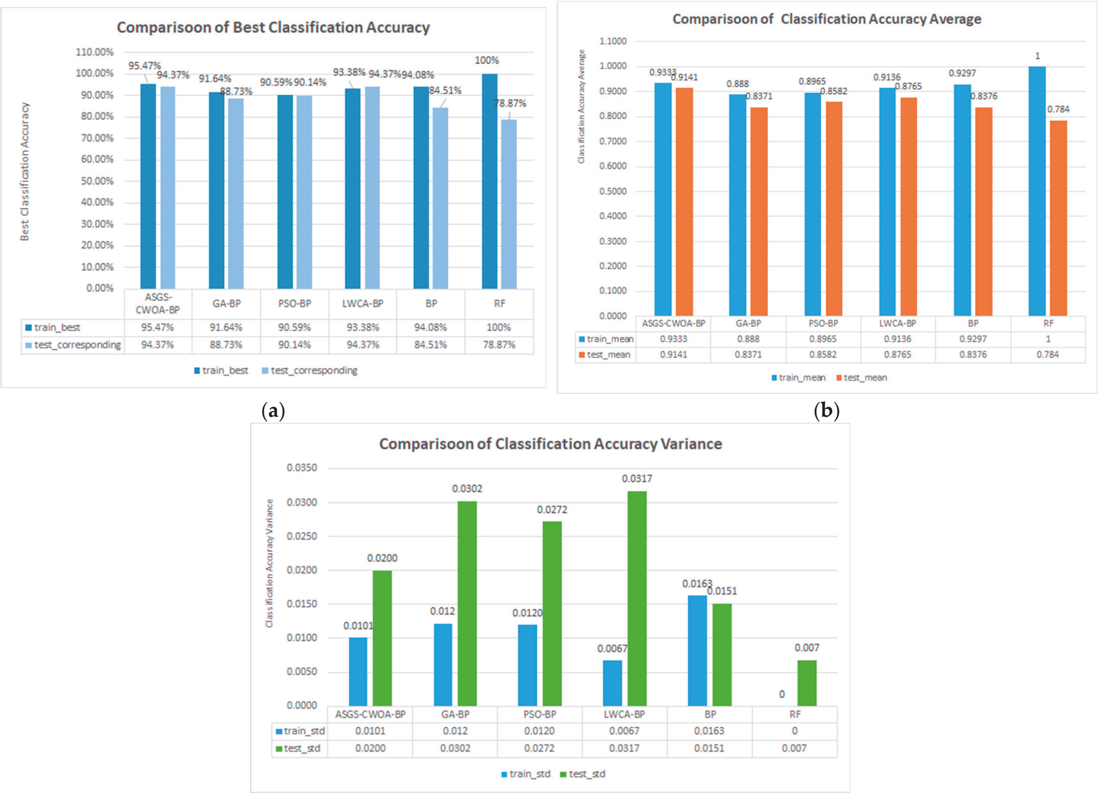
Figure 5. Classification accuracy comparison curve. (**a**) Comparison on the training set; (**b**) comparison on the test set; (**c**) ASGS-CWOA-BP vs. GA-BP; (**d**) ASGS-CWOA-BP vs. PSO-BP; (**e**) ASGS-CWOA-BP vs. LWCA-BP; (**f**) ASGS-CWOA-BP vs. BP; (**g**) ASGS-CWOA-BP vs. RF. Red: ASGS-CWOA-BP; green: GA-BP; black: PSO-BP; blue: LWCA-BP; cyan: original BP based on gradient descent; pink: RF. Circles indicate data from the training set while * indicates data from the test set.

Table 3. Experimental records.

Order	Classification Accuracy on the Training Set						Classification Accuracy on the Test Set					
	ASGS-CWOA-BP	GA-BP	PSO-BP	LWCA-BP	BP	RF	ASGS-CWOA-BP	GA-BP	PSO-BP	LWCA-BP	BP	RF
1	0.9233	0.885	0.9094	0.9094	0.892	1	0.9577	0.8028	0.8732	0.8592	0.7887	0.7746
2	0.9373	0.878	0.892	0.9059	0.892	1	0.9296	0.8873	0.831	0.8732	0.7887	0.7887
3	0.9408	0.885	0.885	0.9094	0.9164	1	0.9014	0.831	0.8732	0.8732	0.831	0.7887
4	0.9373	0.8746	0.9059	0.9164	0.9094	1	0.9014	0.831	0.9014	0.8732	0.831	0.7746
5	0.9338	0.9164	0.9024	0.9199	0.9094	1	0.8873	0.8873	0.8451	0.8592	0.831	0.7746
6	0.9164	0.8711	0.8885	0.9338	0.9094	1	0.8732	0.831	0.8732	0.9437	0.831	0.7887
7	0.9477	0.8955	0.8885	0.9268	0.9094	1	0.9014	0.8451	0.8873	0.9014	0.831	0.7746
8	0.9547	0.892	0.8955	0.9129	0.9094	1	0.9437	0.8028	0.8732	0.8592	0.831	0.7887
9	0.9373	0.9094	0.878	0.9164	0.9094	1	0.9014	0.8592	0.8732	0.8451	0.831	0.7887
10	0.9199	0.892	0.9129	0.9129	0.9338	1	0.9014	0.8592	0.8451	0.9014	0.8592	0.7887
11	0.9477	0.8815	0.899	0.9199	0.9338	1	0.9155	0.8028	0.831	0.9155	0.831	0.7887
12	0.9408	0.8955	0.9164	0.9164	0.9338	1	0.9296	0.8732	0.8873	0.8873	0.831	0.7887
13	0.9338	0.8885	0.8746	0.9059	0.9408	1	0.9155	0.831	0.8028	0.8873	0.8451	0.7746
14	0.9164	0.9024	0.892	0.9024	0.9408	1	0.9014	0.8873	0.8732	0.8732	0.8451	0.7746
15	0.9233	0.8815	0.9164	0.9129	0.9408	1	0.9155	0.8592	0.8873	0.831	0.8451	0.7887
16	0.9164	0.892	0.885	0.9129	0.9408	1	0.9014	0.831	0.8451	0.8592	0.8451	0.7887
17	0.9268	0.9024	0.8955	0.9094	0.9408	1	0.9296	0.8451	0.8873	0.8873	0.8451	0.7887
18	0.9338	0.885	0.9164	0.9094	0.9408	1	0.9577	0.831	0.8732	0.8592	0.8451	0.7887
19	0.9233	0.892	0.9024	0.9129	0.9408	1	0.9155	0.8169	0.831	0.8732	0.8451	0.7746
20	0.9408	0.8955	0.885	0.9059	0.9408	1	0.9014	0.8732	0.8732	0.8169	0.8451	0.7887
21	0.9373	0.8955	0.885	0.9199	0.9408	1	0.9296	0.8451	0.8732	0.9155	0.8451	0.7887
22	0.9338	0.8955	0.8955	0.9199	0.9408	1	0.9296	0.8169	0.8592	0.9014	0.8451	0.7887
23	0.9477	0.892	0.8885	0.9094	0.9408	1	0.9155	0.8028	0.8028	0.9155	0.8451	0.7887
24	0.9268	0.8815	0.899	0.9129	0.9408	1	0.9014	0.8169	0.8169	0.8732	0.8451	0.7887
25	0.9408	0.8711	0.899	0.9164	0.9408	1	0.9296	0.7746	0.8028	0.831	0.8451	0.7887
26	0.9303	0.8711	0.9094	0.9059	0.9408	1	0.9296	0.8028	0.8732	0.8028	0.8451	0.7746
27	0.9233	0.8641	0.9024	0.9164	0.9408	1	0.8873	0.8169	0.8592	0.8732	0.8451	0.7746
28	0.9303	0.8711	0.9094	0.9059	0.9408	1	0.9155	0.8873	0.8592	0.9155	0.8451	0.7887
29	0.9408	0.899	0.878	0.9164	0.9408	1	0.9155	0.8169	0.8592	0.8873	0.8451	0.7887
30	0.9373	0.885	0.8885	0.9129	0.9408	1	0.8873	0.8451	0.8732	0.9014	0.8451	0.7746

In particular, RF was used as a comparison algorithm to analyze the performance of BPs. It can be seen that RF has a very good curve for the training set but a poor one for the test set (Figure 5g), which means that the model trained by RF will be overfitted due to the small dimensions of the feature vectors, which also shows the superiority of the algorithm proposed in this study.

Finally, the best record, average and variance of the classification accuracy are shown in Figure 6a–c, respectively, from which we can see that the proposed method was better for the best value of classification accuracy than any algorithm except RF on the training set, as well as for the average value. However, the variance of ASGS-CWOA-BP was not better than that of LWCA-BP and RF on the training set, while it was better than that of GA-BP, PSO-BP and BP, as shown in Figure 6c. However, the variance of ASGS-CWOA-BP was less than that of GA-BP, PSO-BP and LWCA-BP and was basically the same as that of BP, although it was a little worse than RF. Thus ASGS-CWOA-BP had the best performance of in terms of the best value and high robustness.



(**a**) (**b**)

(**c**)

Figure 6. Statistical analysis of the classification accuracy. (**a**) Best recorded classification accuracy; (**b**) average classification accuracy; (**c**) variance of the classification accuracy.

4. Conclusions

To improve the recognition accuracy of coal gangue images, a coal gangue image recognition method based on the BP neural network and ASGS-CWOA (ASGS-CWOA-BP) was proposed, which makes two key contributions. Firstly, a new feature extraction method regarding the unique features of coal and gangue images is proposed. Additionally, a method using ASGS-CWOA to optimize the parameters of the BP neural network was introduced to address the issue of low accuracy in coal gangue image recognition, and a BP neural network with a simple structure and reduced computational consumption was designed. The theoretical research and experimental results revealed that compared with GA-BP, PSO-BP, LWCA-BP, BP and RF, ASGS-CWOA-BP had the best classification accuracy and high robustness under the same conditions.

Compared with the five other algorithms, ASGS-CWOA-BP performed well in most cases on the training set and test set, and its best classification accuracy on the training set was 95.47% while that on the test set was 94.37%, as shown in Table 4 and Figure 6a. It should be emphasized that this was achieved under extremely limited conditions as follows: (1) the structure of the BP-based recognition network was extremely simple (only six-dimensional feature vectors were required in the input layer and only four nodes in the hidden layer), and (2) the number of samples was very small (only 358 coal gangue image samples). These extremely limited conditions greatly reduced the amount of calculation, and the GPU was not used from beginning to end; therefore, all simulation experiments can be implemented only on a laptop with ordinary performance, as detailed above, which shows that the new method proposed in this article has superior performance. In fact, the recognition model trained by this method is quite suitable for use in mobile portable coal gangue image recognition equipment with weak computing power and low energy consumption.

Table 4. Statistical analysis of classification accuracy.

Method	Training Set			Test Set		
	Best Accuracy	Average	Variance	Corresponding Accuracy	Average	Variance
ASGS-CWOA-BP	95.47%	0.9333	0.0101	94.37%	0.9141	0.02
GA-BP	91.64%	0.888	0.0122	88.73%	0.8371	0.0302
PSO-BP	90.59%	0.8965	0.012	90.14%	0.8582	0.0272
LWCA-BP	93.38%	0.9136	0.0067	94.37%	0.8765	0.0317
BP	94.08%	0.9297	0.0163	84.51%	0.8376	0.0151
RF	100%	1	0	78.87%	0.784	0.0068

However, what needs to be remembered is that the most popular image recognition model based on deep learning has higher and better recognition or classification accuracy and has been studied by a considerable number of scholars. Unfortunately, the network of this technology is complex (with many levels and a large amount of calculation) and often requires a large number of image samples. On the contrary, this is exactly the advantage of the method proposed in this article.

Our future work is to continue to improve the performance of the wolf pack optimization algorithm and to apply it to optimize a more complex BP-based recognition network to increase the feature dimensions of the extracted coal gangue images and increase the number of samples to improve the classification accuracy of coal gangue images.

Author Contributions: D.W. conceived the algorithm framework and wrote the article; J.N. and D.W. performed the program experiments; J.N. and T.D. contributed the data. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by National Key Research and Development Program of China (grant no. SQ2018YFC060172).

Acknowledgments: The authors are grateful to their peer experts for the full support of this paper and thank Zhuhai Xinhe Technology Co., Ltd. and China University of Mining and Technology-Beijing for providing the necessary scientific research environment, as well as special thanks to Beijing Union University for its support of scientific research funds.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Link: https://pan.baidu.com/s/1dlcJGE6_UYNn3vYoQdH_BgExtraction (accessed on 1 February 2022). Code: 4033.

References

1. Qian, M.; Xu, J.; Wang, J. Further on the sustainable mining of coal. *J. China Coal Soc.* **2018**, *43*, 1–13.
2. Zhou, N.; Yao, Y.; Song, W.; He, Z.; Meng, G.; Liu, Y. Present situation and Prospect of coal gangue treatment technology. *J. Min. Saf. Eng.* **2020**, *37*, 11.
3. Xue, G.; Li, X.; Qian, X.; Zhang, Y. Coal-gangue image recognition in fully-mechanized caving face based on random forest. *Ind. Mine Autom.* **2020**, *46*, 57–62.
4. Xu, Z.Q.; Lv, Z.Q.; Wang, W.D.; Zhang, K.; Lv, H. Machine vision recognition method and optimization for intelligent separation of coal and gangue. *J. China Coal Soc.* **2020**, *45*, 2207–2216.
5. Yu, G. Expanded order co-occurrence matrix to differentiate between coal and gangue based on interval grayscale compression. *J. Image Graph.* **2012**, *8*, 966–970.
6. Yu, L. A New Method for Image Recognition of Coal and Coal Gangue. *Mod. Comput.* **2017**, *17*, 68–72.
7. Rao, Z.; Wu, J.; Li, M. Coal-gangue image classification method. *Ind. Mine Autom.* **2020**, *46*, 69–73.
8. Wen, X. *Intelligent Fault Diagnosis Technology: Matlab Application*; Beijing University of Aeronautics and Astronautics Press: Beijing, China, 2015.
9. Zheng, Y.-G.; Wang, P.; Ma, J.; Zhang, H.B. Remote sensing image classification based on BP neural network model. *Trans. Nonferrous Met. Soc. China* **2005**, *15*, 232–235.
10. Chen, Y.X.; Liao, X.D.; Wang, J.H.; Tao, Z.; Sui, L.Y. Small Image Recognition Classification Based on PCA and GA-BP Neural Network. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC 2018), Xi'an, China, 25–27 May 2018; pp. 1360–1363.
11. Zhou, W.C.; Xie, G.S.; Liu, B. The application of mixed GA-BP algorithm on remote sensing image classification. In Proceedings of the Conference: Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Classification of Remote Sensing Images, Guangzhou, China, 28–29 June 2008.
12. Liu, G.; Wei, X.; Zhang, S.; Cai, J.; Liu, S. Analysis of epileptic seizure detection method based on improved genetic algorithm optimization back propagation neural network. *Shengwu Yixue Gongchengxue Zazhi/J. Biomed. Eng.* **2019**, *36*, 24–32.
13. Liu, M.; Guan, W.; Yan, J.; Hu, H. Correlation identification in multimodal weibo via back propagation neural network with genetic algorithm. *J. Vis. Commun. Image Represent.* **2019**, *60*, 312–318. [CrossRef]
14. Yu, J.; Zhang, Z.; Guo, P.; Qin, H.; Zhang, J. Multispectral remote sensing image classification based on PSO-BP considering texture. In Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA), Chongqing, China, 25–27 June 2008; pp. 6803–6806.
15. Chen, Y.X.; Liao, X.D.; Wang, J.H.; Tao, Z.; Sui, L.Y. Small Image Recognition Classification Based on Random Dropout and PSO-BP. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, Xi'an, China, 25–27 May 2018; pp. 1243–1246.
16. Yu, J.; Li, Y.; Zhang, Z.S.; Jiang, J.C. Research on supervised classification of fully polarimetric SAR image using BP neural network trained by PSO. In Proceedings of the World Congress on Intelligent Control and Automation (WCICA), Jinan, China, 7–9 July 2010; pp. 6152–6157.
17. Wei, B.; Hu, L.; Zhang, Y.; Zhang, Y. Parts Classification based on PSO-BP. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC 2020), Chongqing, China, 12–14 June 2020; pp. 1113–1117.
18. Wang, W.; Lu, K.; Wu, Z.; Long, H.; Zhang, J.; Chen, P.; Wang, B. Surface defects classification of hot rolled strip based on improved convolutional neural network. *ISIJ Int.* **2021**, *61*, 1579–1583. [CrossRef]
19. Dong, H.; Jiao, R.; Huang, M. Research on recognition method of cloud precipitation particle shape based on bp neural network. *MATEC Web Conf.* **2021**, *336*, 06011. [CrossRef]
20. Yang, C.; Tu, X.; Chen, J. Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search. In Proceedings of the International Conference on Intelligent Pervasive Computing, Jeju, Korea, 11–13 October 2007; Volume 871, pp. 462–467.
21. Li, H.; Wu, H. An oppositional wolf pack algorithm for Parameter identification of the chaotic systems. *Opt. Int. J. Light Electron Opt.* **2016**, *127*, 9853–9864. [CrossRef]
22. Chen, Y.B.; Mei, Y.S.; Yu, J.Q.; Su, X.L.; Xu, N. Three-dimensional Unmanned Aerial Vehicle Path Planning Using Modified Wolf Pack Search Algorithm. *Neurocomputing* **2017**, *266*, 445–457.
23. Yang, N.; Guo, D.L. Solving Polynomial Equation Roots Based on Wolves Algorithm. *Sci. Technol. Vis.* **2016**, *15*, 35–36.
24. Zhou, Q.; Zhou, Y. Wolf colony search algorithm based on leader strategy. *Appl. Res. Comput.* **2013**, *30*, 2629–2632.
25. Wang, D.; Ban, X.; Qian, X. An Adaptive Shrinking Grid Search Chaos Wolf Optimization Algorithm with Adaptive Standard-Deviation Updating Amount. *Comput. Intell. Neurosci.* **2020**, *2020*, 7986982. [CrossRef] [PubMed]

symmetry

MDPI

Article

DGAN-KPN: Deep Generative Adversarial Network and Kernel Prediction Network for Denoising MC Renderings

Ahmed Mustafa Taha Alzbier [1,2] **and Chunyi Chen** [1,*]

[1] School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China; amt4047@gmail.com
[2] School of Computer and Information Technology, Omdurman Islamic University, Omdurman 382, Sudan
* Correspondence: chenchunyi@cust.edu.cn

Abstract: In this paper, we present a denoising network composed of a kernel prediction network and a deep generative adversarial network to construct an end-to-end overall network structure. The network structure consists of three parts: the Kernel Prediction Network (KPN), the Deep Generation Adversarial Network (DGAN), and the image reconstruction model. The kernel prediction network model takes the auxiliary feature information image as the input, passes through the source information encoder, the feature information encoder, and the kernel predictor, and finally generates a prediction kernel for each pixel. The generated adversarial network model is divided into two parts: the generator model and the multiscale discriminator model. The generator model takes the noisy Monte Carlo-rendered image as the input, passes through the symmetric encoder–decoder structure and the residual block structure, and finally outputs the rendered image with preliminary denoising. Then, the prediction kernel and the preliminarily denoised rendered image is sent to the image reconstruction model for reconstruction, and the prediction kernel is applied to the preliminarily denoised rendered image to obtain a preliminarily reconstructed result image. To further improve the quality of the result and to be more robust, the initially reconstructed rendered image undergoes four iterations of filtering for further denoising. Finally, after four iterations of the image reconstruction model, the final denoised image is presented as the output. This denoised image is applied to the loss function. We compared the results from our approach with state-of-the-art results by using the structural similarity index (SSIM) values and peak signal-to-noise ratio (PSNR) values, and we reported a better performance.

Keywords: deep learning; generative adversarial network; kernel prediction network; Monte Carlo rendering; auxiliary features; high frequency

Citation: Alzbier, A.M.T.; Chen, C. DGAN-KPN: Deep Generative Adversarial Network and Kernel Prediction Network for Denoising MC Renderings. *Symmetry* **2022**, *14*, 395. https://doi.org/10.3390/sym14020395

Academic Editor: Antonio Palacios

Received: 5 January 2022
Accepted: 10 February 2022
Published: 16 February 2022

1. Introduction

Due to the continuous development of deep learning methods in recent years, there have been many works using deep learning to denoise ordinary images; therefore, the convolution neural network has been widely used in the research of image denoising. Pathak et al. [1] used an encoder to encode and trained to generate images conditioned on context, in which the encoders learn a representation that is competitive with other models trained with auxiliary supervision, which captures the appearance, the semantics of visual structures, and complete image restoration. Bert et al. [2] showed a dynamic parameter network structure in which the parameters of the kernel are dynamically adjusted according to the input, because it has high flexibility and avoids a large number of increases, with the condition of the model parameters. Bako et al. [3] proposed a denoising algorithm based on convolutional neural networks, which decomposes the image into diffuse reflection and specular reflection. Therefore, the two parts are trained separately. In addition, for image effects that are not reflected in the input features or included in the training data, the results after denoising will appear blurry. Further, using a fixed filter solves the drawback, but the method is still dependent on filtering kernels in a wide range and becomes acceptably field-limited.

Vogels et al. [4] proposed another denoising network structure based on kernel prediction. In this work, they showed three network structures that can be used in different situations. The first network for a single frame of the input image has four parts: a source encoder, a spatial feature extractor, a kernel predictor, and weight reconstruction. The spatial feature extractor includes multiple residual network structure blocks [5]. The second and third network structures are temporal denoisers for multi-frame and multi-standard images. Each frame is first passed through a separate original encoder and spatial feature extractor, and then combined and inputted to the temporal feature extractor and kernel predictor to obtain the final denoised image.

Recently, Mildenhall et al. [6] also proposed a denoising process for images taken with a camera held by a hand, and then using a convolutional neural network structure. This network structure can learn to obtain a kernel that follows spatial changes; the kernel can denoise or register the image, and the trained network has a good denoising effect on most noisy images. Mao et al. [7] proposed a novel deep self-encoding network structure for image restoration. This network has an encoder and a decoder. The encoder and decoder are symmetrical in structure, with convolutional and reversed layers, respectively. In particular, this network also uses skip connections, which can effectively solve the difficulty in deep network training and the problem of gradient disappearance, and can, at the same time, transfer image detail information from the convolutional layer to the deconvolution. The layering helps to build a clear real image. The above work has proven that deep learning has very good performance in image restoration and image denoising.

At present, the use of deep learning methods to denoise the Monte Carlo-rendered images has gradually attracted attention. Unlike general image restoration, in the process of denoising the Monte Carlo-rendered image, in addition to the color information of the pixels, additional auxiliary information can be used, such as depth, normal, and albedo.

The Monte Carlo denoising method of the joint kernel prediction network and generation adversarial network proposed in this paper uses the generation adversarial network to perform preliminary denoising on the Monte Carlo-rendered image, and then applies the prediction kernel output by the kernel prediction network to the preliminary denoising image to obtain the final result. The difference between the method in this paper and the existing kernel prediction Monte Carlo denoising method is reflected in the following aspects [3,4,8]. First, as an improvement of the kernel prediction network method, this paper introduces an adversarial generation network to generate preliminary denoising results and denoise on this basis, instead of directly applying the prediction kernel to the original noisy rendered image. For support, the two work together to assemble the two into an end-to-end denoising network for joint training. Secondly, a loss function is added to support collaborative training of the kernel prediction network and the generative adversarial network to improve the scene detail retention ability and the scene clarity and contrast.

The denoising network in this article is used to process input images with a low sampling rate such as 4 spp, and it can obtain better results. In addition, when constructing the dataset, this article deliberately uses multiple renderers to generate supervised data, which effectively improves the generalization ability of the network. The kernel prediction network in our model takes the auxiliary information images of the Monte Carlo rendering as the input, and the adversarial generation network takes the noisy rendered image itself as the input. This processing method can ensure that each part of the network can encode more image features, thereby capturing more scene details.

In order to further solve the problems of the above two methods and improve the denoising effect of Monte Carlo-rendered images, the main contributions of this paper are the following three points:

- In the first part of this paper, we propose a new end-to-end Monte Carlo denoising rendered image based on the deep learning network structure, and we use the kernel prediction network to optimize the generalization ability of the denoising method for better scene structure and detail retention capabilities.

- We introduce a loss function based on adversarial training to make network training more stable and effective, to improve the clarity and contrast of the denoised image, and to retain more image details.
- We prove that a few auxiliary features can improve the noise reduction effect and solve the loss of high-frequency details of our approach to some extent.
- Our approach is applied to the deep convolutional neural network and makes the learning ability of the network more powerful, with less time-consuming processing.

2. Related Work

In recent years, the Generative Adversarial Network (GAN) [9] has also been shown to achieve good results in image restoration and high-resolution image generation [10–14]. Moreover, generative adversarial networks have also played a role in image denoising works [15]. Regarding the problem of image denoising by Monte Carlo rendering, in 2019, Xu et al. [16] found that the recent Monte Carlo denoising method based on deep learning is more dependent on artificial optimization goals. Therefore, they proposed a method to denoise the Monte Carlo-rendered image by introducing a generative adversarial network. The network then processed the highlights and diffuse components in the rendered image. Finally, the denoised image was output directly and excellent results were obtained. Therefore, the generative adversarial network has considerable potential in the problem of Monte Carlo-rendered image denoising. Unlike the work of Xu et al. [16], Monte Carlo denoising is based on the kernel prediction network, and the generative adversarial network is integrated into the kernel prediction network as a preliminary denoising generation model. In 2019, Xin et al. [17] extracted structure and texture details from auxiliary features in the rendering stage. Then, they used a fusion sub-network to obtain the details map, and finally used the dual-encoder network to denoise MC renderings. However, this method consumes processing time. Ghrabi et al. [8] proposed a network structure with permutation invariance, used a multilayer coding structure to encode sample data to obtain the splat kernel, and then used this check to reconstruct the input image, making their method the best state-of-the-art method that uses the kernel prediction network and is based on samples. Unfortunately, increasing the number of samples increases the time consumption of the method. In 2020, Munkberg et al. [18] suggested extracting the compressed information representation of each sample by separating the sample into a fixed number of sections, called layers. Through a data-based method, this method learns the unique kernel weight of each pixel in each layer and how to filter the composite layer. This adjustment enables the degreaser operation to achieve a good trade-off between cost and quality. In addition, it provides an effective way to control performance and memory properties, because the algorithm table is the number of layers rather than the number of samples. Moreover, via the separation of two-layer samples, the denoiser achieves an interaction rate and produces an image quality similar to that of the larger network.

Again in 2020, Yifan et al. [19] proposed the Adversarial Denoising for MC Renderings network, which used many convoluted dense blocks to extract rich information of auxiliary buffers, and then used these various hierarchical features to modify the noisy features in the residual blocks. Furthermore, they presented the channel mechanism and spatial interest to exploit property dependencies between channels and spatial features.

In 2021, Yu et al. [20] modified the standard self-attention mechanism to the auxiliary feature guided self-attention module to denoising Monte Carlo rendering based on a deep learning network, which effectively involves the complex denoising process.

Generally, Monte Carlo-rendered image denoising based on deep learning is mainly divided into two categories. One is based on methods of the kernel prediction network [3,4], which uses network estimation to generate a prediction kernel and applies this prediction kernel to the noise input to obtain the final denoised image. The other is to directly map the noise input to the high-quality rendered images; the network is used to directly generate the final denoising rendered image. Thus, the key idea of this paper is to combine the strategies of these two methods to build a Monte Carlo-rendered image denoising model. This is because the method based on kernel prediction is effective in the restoration and preservation of scene structure and scene details. However, the adaptability is poor when the denoised renderer of the image is different from the renderer used in the training set, but the network that directly outputs the denoising results will have relatively good generalization ability.

This paper proposes a method to generate realistic rendered images using an end-to-end network structure. First, the renderer is used to render the 3D model at a low sampling per pixel to obtain a low-resolution image. Therefore, the rendering time is relatively short; then, the proposed new image denoising network is used to obtain a high-quality image.

3. The Method

3.1. Model Architecture

In this paper, we propose a new network structure based on the kernel prediction network and the Deep Generation Adversarial Network (DGAN) to build this function. The kernel prediction model alone or the DGAN model with noise input can be used to generate denoised rendered images [3,4,16]. The difference between these two models is that the kernel prediction network first learns the prediction kernel from the input data and then applies the prediction kernel to the pixels of the noise image. The DGAN learns the connection between the noise pixel and the real pixel as the target, and maps the noise pixel to the reference real pixel, thereby directly generating a denoised image close to the real image. The kernel prediction network can restore the scene structure well and retain the details of the scene, and the DGAN-based method can have better generalization ability. The characteristics of these two models inspired this paper to combine the two to obtain a better denoising effect promotion and generalization ability improvement.

Generally, the idea of combining these two models is not complicated, but we have made many improvements to make these networks work together: First, we improved the previous kernel prediction network [3,6,21] and improved the feature encoder, making it have a better ability to capture scene details and have a better adaptability to input data from different renderers. Secondly, the DGAN network structure contains 4 discriminant networks of different scales as discriminators to supervise the encoding of details at different scales. In addition, to improve the reconstruction quality, the result after the prediction kernel reconstruction is used as a new noise image and the network is used again for the second denoising. This process is repeated many times to obtain the final denoised image. Finally, we propose adding a loss function to the network, which can be trained stably while improving the detail retention ability of the denoising results, the sharpness, and the contrast in the final image.

In addition, the loss function must accurately capture the difference between the estimated pixel value and the real pixel value, and it is easy to adjust and optimize. In Section 3.1.4, we introduce the proposed loss function. Finally, to avoid overfitting in our network, we made a dataset that contains a large amount of data. It takes a lot of time and computational cost to make a dataset that contains a large number of real images, noisy images, and auxiliary features.

Figure 1. This network structure consists of three parts: the deep generation adversarial network model, kernel prediction network model, and image reconstruction model.

3.1.1. Deep Generation Adversarial Network (DGAN)

For a deep-generation adversarial network, the generator is divided into three parts, the first part being the encoder. The encoder contains 4 convolutional layers, and each convolutional layer contains convolution, and three operations of instance normalization and ReLu activation. After the encoder, there are several residual blocks and a structure that combines the input and output information [5]. The specific structure of a residual block contains 2 convolutional layers, and each convolutional layer has 512 convolution kernels with a size of 3 × 3 and a stride size of 1; similarly, each convolutional layer is composed of four parts: convolution, instance normalization, and ReLu activation. The residual block introduces a skip connection by adding between the convolutional layers. Then, the decoding part is similar to the encoding part. Therefore, the encoding part is upsampling after the output of the fourth convolutional layer and the eighth convolutional layer, and the output of the fourth convolutional layer and the fourth convolutional layer of the decoding part is jointly upsampled through a skip connection. Then, they are combined after upsampling.

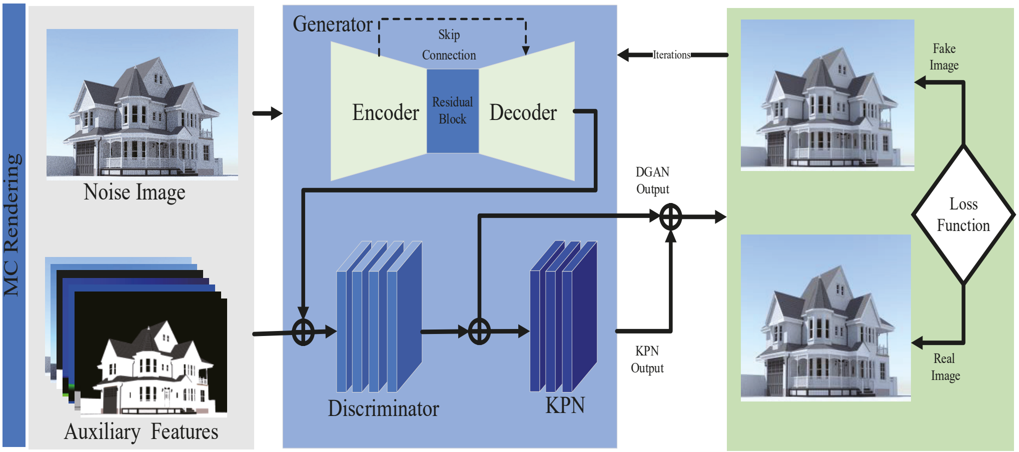


Figure 1. The overall structure of the proposed method.

To simplify the description in this section, a network layer composed of these three operations is collectively referred to as a convolutional layer. The first convolutional layer contains 64 convolution kernels, the number of output channels is 64, and for each convolution kernel, the size is 3 × 3 and the stride size is 2. Similarly, the number of convolution kernels of the second convolution layer and the third convolution layer is 128, 256, and 512, respectively, the size of the convolution kernel is constant 3 × 3, and the stride size is 2.

3.1.2. The Kernel Prediction Network (KPN)

The difference between the kernel prediction network (KPN) and the general method of denoising using neural networks is that the kernel prediction network does not directly output a denoised image, but the kernel predictor estimates a filter kernel of size $k \times k$ for each pixel of the noise image, where, in the implementation of this article, $k = 19$. The kernel predictor contains three convolutional layers, each convolutional layer is filled with zeros, the size of the kernel is 1 × 1 convolution kernel, the stride size is 1, and the number of output channels of each convolutional layer is $19 \times 19 = 361$. These prediction kernels enter the reconstruction model and the denoising structure of DGAN to generate clean images.

As different input images may be rendered by different renderers or rendering systems, and thus obtained by different samplers or calculation methods, these inputs are likely to have different noise characteristics, and then the network structure must have applicability to these different inputs [8]. As the first part of the encoder, it is proposed to make the network have this applicability, by extracting relatively low-level and common features in the input information to unify complex input information and reduce the impact of different inputs.

Enlarging the size of the convolution kernel helps expand the perceptual domain to obtain more details about the neighborhood information. The output information obtained after the input information passes through these 2 convolutional layers is compared with the original input information through the skip structure as the final output. The introduction of residual blocks in the denoising processing of Monte Carlo-rendered images has been successful in related research work [22]. It has two advantages: First, as the input image is noisy, with many missing pixels and wrong pixel values, the image is very sparse. Therefore, the input is combined before and after through the residual block to obtain more feature information. Second, the residual block can effectively solve the problem of gradient disappearance caused by the excessive depth of the network during the training process, and the convergence of the loss function during the training process can be faster and more stable.

3.1.3. Image Reconstruction

Recall that the pre-denoising image output of the generation network is $\hat{z}$, the kernel obtained by KPN $K = \{k_p, p \in z\}$, where k_P is a $k \times k$ matrix, and its x row and column elements y are marked as $k_p(x, y)$. To ensure that the weight range of each kernel falls in the interval [0,1], and the sum is equal to 1 [21], we first use the SoftMax function to normalize:

$$\hat{K}_P(x,y) = \frac{K_p(x,y)}{\frac{1}{|S|}\sum_{1\le s,t\le k}\exp\left(K_p(x,y)\right)} \tag{1}$$

The meaning of each element in the prediction kernel $\hat{K}_P$ is the degree of influence of each pixel area in the domain $k \times k$ around the pixel p. S is the set of kernel sizes. Therefore, the final reconstructed image can be calculated as follows [21]:

$$\acute{z}(p) = \frac{1}{|S|}\sum_{1\le s,t\le k} \hat{K}_P(x,y)\acute{z}(p+(x,y)) \tag{2}$$

Through weights normalizing, we can estimate the final color value included in each pixel area in the image, which can greatly reduce the search space of the output estimation value in the denoising process, and avoid phenomena such as the color shift effect. Secondly, normalization can also make the gradient of the weight value relatively stable, avoiding large gradient oscillations caused by the high-dynamic-range characteristics of the input image during the training process.

3.1.4. Loss Function Design

The network proposed in this article is made up of KPN and DGAN. Thus, designing a reasonable loss function is a very important issue that enables these two networks to work together and improve the quality of denoising. Specifically, our loss function consists of three parts.

Generate the loss function L_{DGAN}: The generator is responsible for using the input noise image to generate a preliminary denoising rendered image, and the discriminator is responsible for comparing the generated image with the real image. Our dataset is $M = \{m_i = (x_i, f_i), g_i : i = 1, 2, \ldots, N\}$, where R is the number of denoise elimination iterations. We set it in our work as 4 iterations, and then considering the generator as G, and the discriminator is set as $D = D_i,\ i = \{1, 2, 3, 4\}$. The training process for generating an adversarial network is a process of optimizing the loss function L_{DGAN}, such as [23]:

$$min_G max_D L_{DGAN}(G, D) \tag{3}$$

$$min_G max_D \sum_{i=1}^{R} L_{DGAN}(G, D_i) \tag{4}$$

During this, the optimization function (Equation (4)) in the optimization process is used to solve the parameter value of each D_i. Thus, L_{DGAN} reaches the maximum, this D_i is fixed, and then we solve for G to minimize $L_{GAN}(G, D_i)$.

$$G^* = min_G max_D \sum_{i=1}^{R} L_{DGAN}(G, D_i) \tag{5}$$

The generator G^* at this time has the model parameters to produce a reasonable denoised image by Equation (5). On the contrary, we adopt a different general discriminator form [15]. As for the loss function $L_{DGAN}(G, D_i)$ of a single discriminator, instead of letting the discriminator output a probability value to judge the true or false of the sample, $L1$ is used to measure the loss between the two samples, namely [15]:

$$L_{DGAN}(G, D_i) = E\left[\frac{1}{|z|}\sum\nolimits_{p\in z}\|D_i(z(p), g(p)) - D_i(z(p), G(p))\|_1\right] \quad (6)$$

Among them, $|z|$ is the total number of pixels in the image, $D_i(z(p), g(p))$ represents the pixel value of the input image $z(p)$, and the corresponding real image pixel $g(p)$ is the output obtained as the input of the $i\ th$ discriminator. The same principle $D_i(z(p), G(z(p)))$ represents the output obtained by taking the generator output $G(z(p))$ corresponding to $z(p)$ as the input of the $i\ th$ discriminator. E represents the mathematical expectation, which is the average calculation of the loss values calculated for all samples in the dataset.

Equation (6) is only used when training a single adversarial generation network. When KPN and DGAN are trained together, Equation (6) becomes the following form:

$$L_{DGAN}(G, D_i) = E\left[\frac{1}{|z|}\sum\nolimits_{p\in z}\|D_i(z(p), g(p)) - D_i(z(p), \acute{z}(p))\|_1\right] \quad (7)$$

The output of the generator $G(z(p))$ becomes the estimated value of the pixel after the prediction kernel obtained by KPN is applied to the output of the generator $\acute{z}(p)$.

In kernel prediction loss function L_K, the true value of the prediction kernel cannot be obtained, because there is no such label in the dataset. Thus, we use real images g_i for supervision and, at the same time, make the two networks work together. Therefore, L_K is defined as:

$$L_{kernel} = \sum\nolimits_{z_i}\frac{1}{|z_i|}\sum\nolimits_{p\in z}\|\acute{z}(p) - g(p)\|_1 \quad (8)$$

Some state-of-the-art studies [16,24,25] found that comparing $L1$ loss with $L2$ loss can also reduce speckle noise-like artifacts in the reconstructed image, because $L1$ is more sensitive to outliers, such as brighter highlights, which have a great influence on error. Compared with $L1$ loss, $L2$ loss will be more robust to outliers, which is also confirmed in previous literature. However, $L1$ loss or $L2$ loss usually obtained a higher peak signal-to-noise ratio (PSNR) [26], but the result of the blurring of high-frequency components led to a blurry texture. Therefore, it is necessary to adopt other loss functions to compensate for the high-frequency details. Therefore, we add tone loss function L_T. To make the generated denoised image details have better definition, have a better denoising effect on low-contrast and darker noisy images, and improve its contrast, it is subject to the method inspired by [27], added as a new loss function item to improve the denoising effect of the image. L_T has the following form:

$$L_{Tone} = \sum\nolimits_{z_i}\frac{1}{|z_i|}\sum\nolimits_{p\in c}\left\|\frac{\acute{z}(p)}{1+\acute{z}(p)} - \frac{g(p)}{1+g(p)}\right\|_1 \quad (9)$$

Equation (9) is inspired by tone mapping, which can map the pixels in the image from a small range to a larger range so that the picture can be clearer and brighter. It is a common method in image processing. This penalty item can improve the contrast and clarity of the scene. Finally, the overall loss function L_{total} defines a mixture of the above three terms:

$$L_{total} = \alpha L_{DGAN} + \beta L_{kernel} + \omega L_{Tone} \quad (10)$$

Among them, α, β, ω, we set the balance parameters as 0.003, 0.008, and 0.09, respectively. Typically, by using such a loss function to make the overall network structure work together, it becomes an end-to-end overall structure.

Finally, this article chooses the gradient magnitude similarity deviation as the image noise estimate because it is relative to other indicators such as the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [21], because it has achieved good results in public databases for image quality evaluation and the calculation speed is relatively fast.

3.2. Auxiliary Feature

We use the Monte Carlo path-tracing algorithm to render a 3D model, and each pixel needs to shoot a ray from the camera. Then, it records the information when the ray tracer intersects the 3D model for the first time and saves it in the geometry buffer. The saved information includes the texture and material information such as the surface normal, world coordinates, and reflection coefficient of the patch where the intersection point is located, as well as the position of the point in the world coordinate system and the visibility of direct light. This paper does not record information related to a specific scene, such as the position of the light source, intensity, and other attributes of the scene.

The MC images may differ when compared to the ground-truth images, which are clearer and higher-resolution compared to the latter. These differences in training and test data can lead to discrepancies in the actual models. Therefore, it is essential to have datasets that have consistent auxiliary feature images.

Figure 2. The auxiliary feature images include surface normal features (3 channels), RGB color features (3 channels), world position features (3 channels), texture value1 features (3 channels), texture value2 features (3 channels), and the depth feature (1 channel), which contain 13 channels in total, such as the following:

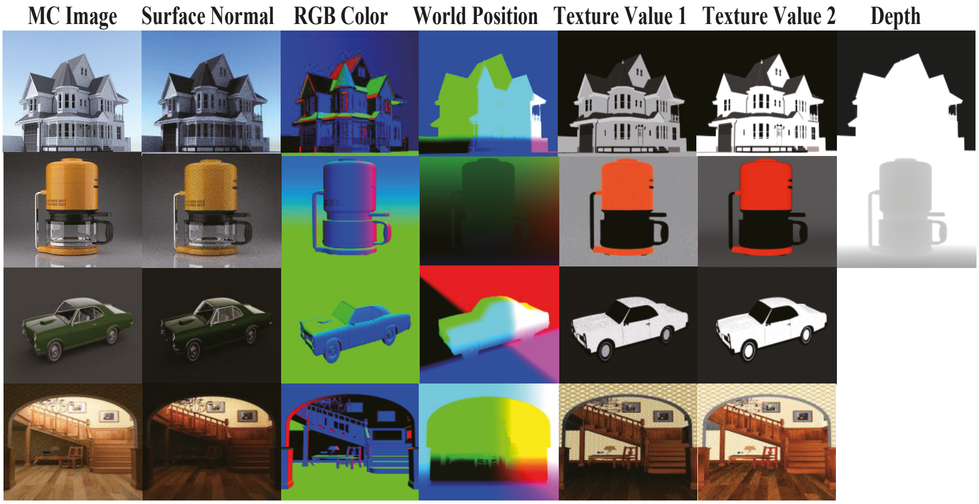


Figure 2. The auxiliary features are rendered with 4 spp; note that in some scenes the depth feature is only white color.

3.3. Dataset and Training

We faced a challenge in the rarity of image datasets, which is hard to find because of the proprietary values. Therefore, crowdsourcing images can be used to address this challenge, and one of the publicly available datasets is the PBRT dataset [28].

Continuous training of the model involves a large and effective dataset. The training dataset of the state-of-the-art is not public. Therefore, we preferred a reasonably interesting dataset consisting of 21 curated scenes available for use with PBRT [28], which can represent different types of scenes and then modify the environment maps and camera parameters. Therefore, the dataset provides complex scenes that are rendered with 4096 spp such as:

Figure 3 shows examples of reference images rendered with tungsten, and this process is time-intensive, up to several days for some scenes.

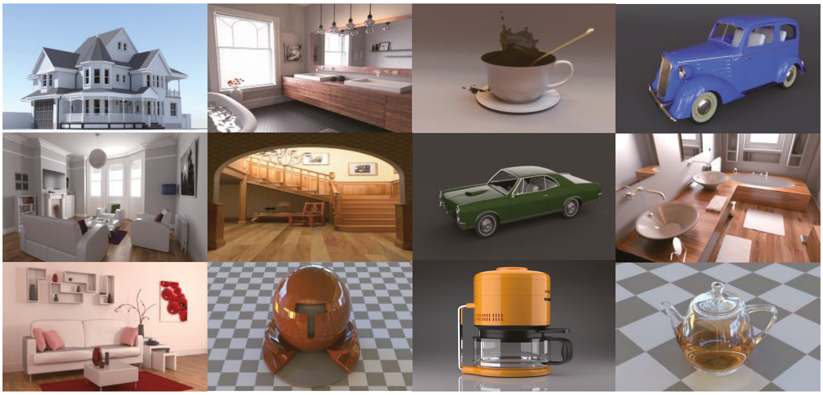


Figure 3. Example of our dataset reference images with 4096 spp.

In contrast, we divide the training process into two phases. First, the DGAN is trained, including the generator model and the multiscale discriminator model. During training, the standard method in [9] is used to optimize the setting of training parameters, the loss function uses Equation (6) and uses the ADAM optimizer [29], and the parameter settings of the remaining optimizer follow [30] to set the recommended parameters. The initial learning rate is set to 0.0001, the learning rate is fixed before the first 200 epochs, and then the learning rate is gradually reduced according to the linear method. The parameter initialization of the network is initialized with a Gaussian distribution with a mean of 0 and a standard deviation of 0.002, and the batch size is set to 1. Each training iteration will randomly disturb the order of the dataset. Then, KPN is trained, Equation (8) is used as the loss function, and the obtained prediction kernel is applied to the noisy RGB image, rather than against the image generated by the generation network. Using this method makes the training of the kernel prediction network stable. The weights and parameters of the kernel prediction network are initialized using the Xavier method [31], and the bias term is set to 0. The ADAM optimizer is also used, and the parameter and learning rate settings of the optimizer are the same as the province training settings of the generator network model.

The second training process of the overall network structure uses the two networks that have been initially trained. In this training, the number of multiple iterations is set to 4, where the result of the image reconstruction model will be denoised again and then after these repetitions to obtain the final denoising result.

4. Results

This paper proposed using a kernel prediction network and generative adversarial network to construct an end-to-end general denoising network structure, as shown in Figure 1. Our network structure consisted of three parts: the kernel prediction network module, generation adversarial network module, and image reconstruction module. The kernel prediction network module takes the auxiliary feature information image as the input, passes through the source information encoder, the feature information encoder, and the kernel predictor, and finally generates a prediction kernel for each pixel.

The generated adversarial network module is divided into two parts: the generator module and the multiscale discriminator module. The generator module takes the noisy Monte Carlo-rendered image as the input, passes through the symmetric encoder–decoder structure and the residual block structure, and finally outputs the rendered image with preliminary denoising. Then, the prediction kernel and the preliminarily denoised rendered image are sent to the image reconstruction module for reconstruction, and the prediction kernel is applied to the preliminarily denoised rendered image to obtain a preliminarily reconstructed rendered image. To further improve the quality of the result and to be more robust, the initially reconstructed rendered image undergoes four iterations of filtering for further denoising, and the final denoised image is obtained after four iterations of the image reconstruction module as outputs. Finally, this denoised image is applied to the loss function.

We evaluated the denoising MC renderings based on the KPN-DGAN method to solve the MC noise image problem and the high-frequency detail loss.

The PSNR and SSIM matrices were used as the quantitative indicators of denoising results. Thus, PSNR calculated the reconstruction error between the denoised and real images based on the mean square sum (MSE). Note that the errors of these matrices are sensitive to noise; as long as a certain pixel value changes and regardless of which direction it changes, the PSNR will also change. Thus, the value range of PSNR is not fixed, and the maximum value is related to the image resolution.

Then, we selected the most representative methods in Monte Carlo image denoising in recent years to compare with our experimental results, which are the KPCN work in 2017 [3], AMCD, and DEMC in 2019 [16,17]. Note that the selected scene uses the 4 spp noise image rendered by the tungsten renderer [32]. The results are as follows:

Figure 4 shows the ablation experiment of this paper, and the enlarged area of image details the MC-rendered image with 4 spp, our result against the AMCD result, KPCN result, DEMC result, and the reference rendered image with 4096 spp. The effect of our approach is better in the final denoising result in terms of subjective details and objective indicators, such as the radiator details and the geometrical objects reflecting sharper on a lamp of the automobile scene, maintaining the barrier shape that does not overlap and the lines in the house scene. In the livingroom2 scene, our method performance is also better and enhances sharp edges with greater detail, unlike other methods. The effect of AMCD algorithms is good, but most of the results are a little blurred. The DEMC and KPCN are poor, because the results have many stains. Generally, comparing the results showed that our approach is better at denoising the MC-rendered image, while retaining and restoring the details and structure of the scene.

The PSNR and SSIM index values are reported under each image, and higher values indicate a better result. The network of our approach performed well and reduced the time consumption of denoising. Therefore, we compared our method against prior methods, with similar processing conditions and an equal sample for all methods, and the results of the average SSIM and PSNR scores are as follows:

In order to further observe the results of the method in this paper, more scene models were selected for comparison; we highlighted the difference between diffuse and specular components, and the relationship to high-frequency details. Thus, the following comparison experiments were compared with AMCD, the DEMC work in 2019 [16,17], and the AFGSA work in 2021 [20]; all of these techniques have public released codes and weights. The 3D model is still the tungsten renderer, and the sampling rate is 4 spp. The experimental results are as follows:

Usually, the specular and diffuse components have different noise patterns and are highly dependent upon the smoothness or texture of the surface properties. Figure 5 shows the other methods that led to unsatisfactory results, with disturbing effects on material and glass, such as blur region, glossy reflections, depth of field, area lighting, and global illumination. Therefore, they need to take advantage of auxiliary features in different ways. The material scene showed an erroneous texture, the reflected illumination was poor, the teapot scene had blurred details and was smoother, and there was a glow reflection in the glass with more noise in the coffee scene. All methods accepted our approach.

In all noise reduction tests, our method always performed better than several state-of-the-art solutions. Tables 1 and 2 show the SSIM and PSNR values and time process for all noise reduction results. Accordingly, our method consistently had smaller errors, with higher SSIM values and less time consumption than state-of-the-art methods.

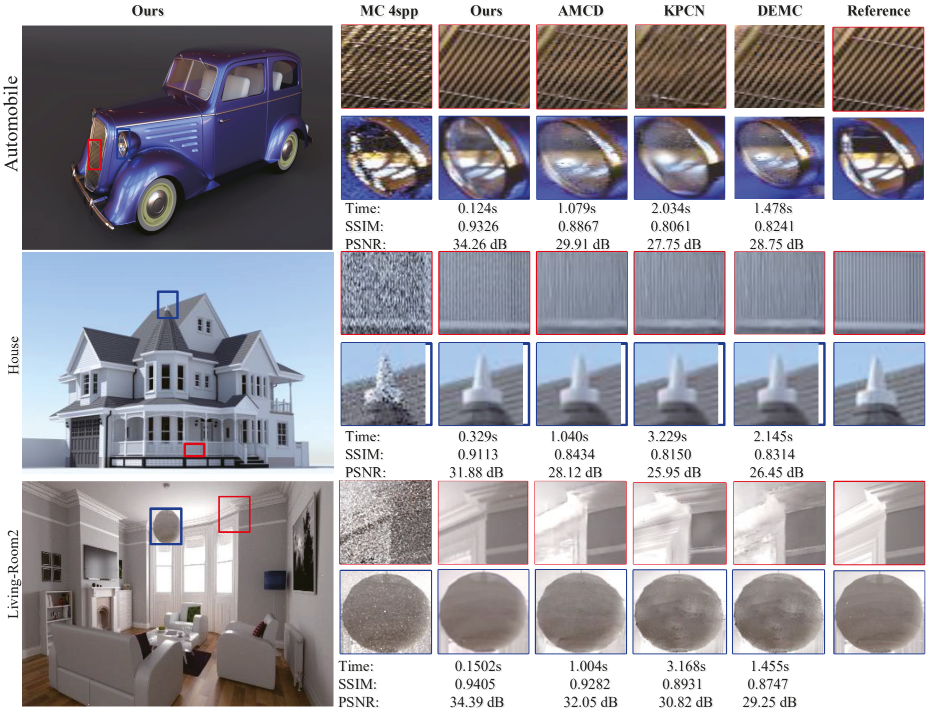

Ours
MC 4spp
Ours
AMCD
KPCN
DEMC
Reference
Automobile
Time: 0.124s 1.079s 2.034s 1.478s
SSIM: 0.9326 0.8867 0.8061 0.8241
PSNR: 34.26 dB 29.91 dB 27.75 dB 28.75 dB
House
Time: 0.329s 1.040s 3.229s 2.145s
SSIM: 0.9113 0.8434 0.8150 0.8314
PSNR: 31.88 dB 28.12 dB 25.95 dB 26.45 dB
Living-Room2
Time: 0.1502s 1.004s 3.168s 1.455s
SSIM: 0.9405 0.9282 0.8931 0.8747
PSNR: 34.39 dB 32.05 dB 30.82 dB 29.25 dB

Figure 4. The comparison of the results of this paper with AMCD, KPCN, and DEMC.

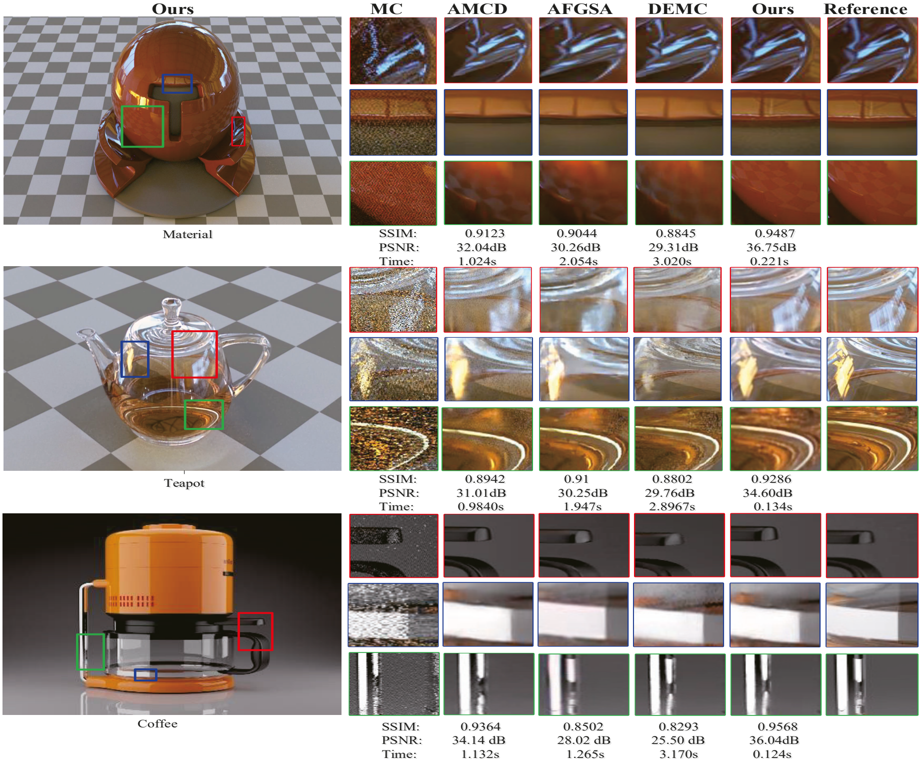
Table 1. The SSIM, PSNR values, and time process results of our approach against the AMCD, KPCN, and DEMC results.

Scene	Ours			AMCD			KPCN			DEMC		
	SSIM	**PSNR**	**Time(s)**	**SSIM**	**PSNR**	**Time(s)**	**SSIM**	**PSNR**	**Time(s)**	**SSIM**	**PSNR**	**Time(s)**
Automobile	0.9326	34.26	0.124	0.8867	29.91	1.079	0.8061	27.75	2.034	0.8241	28.75	1.478
House	0.9113	31.88	0.329	0.8434	28.12	1.04	0.815	25.95	3.229	0.8314	26.45	2.145
Living-room2	0.9405	34.39	0.1502	0.9282	32.05	1.004	0.8931	30.82	3.168	0.8747	29.25	1.455

Table 2. The SSIM, PSNR values, and time process results of our approach against the AMCD, DEMC, and AFGSA results.

Scene	Ours			AMCD			AFGSA			DEMC		
	SSIM	**PSNR**	**Time(s)**	**SSIM**	**PSNR**	**Time(s)**	**SSIM**	**PSNR**	**Time(s)**	**SSIM**	**PSNR**	**Time(s)**
Material	0.9487	36.75	0.221	0.9123	32.04	1.024	0.9044	30.26	2.054	0.8845	29.31	3.020
Teapot	0.9286	34.60	0.134	0. 910	31.01	0.984	0.902	30.76	1.947	0.8942	29.25	2.867
Coffee	0.9568	36.04	0.124	0.9364	34.14	1.133	0.8502	28.02	1.265	0.8293	25.50	3.170

Finally, the KPN-DGAN denoised the Monte Carlo-rendered image with the auxiliary features, which reduced the image noise with a low samples rate, and restored the scene structure details, to improve the quality of rendered images with less time-consuming processing.



Material

	AMCD	AFGSA	DEMC	Ours
SSIM:	0.9123	0.9044	0.8845	0.9487
PSNR:	32.04dB	30.26dB	29.31dB	36.75dB
Time:	1.024s	2.054s	3.020s	0.221s

Teapot

	AMCD	AFGSA	DEMC	Ours
SSIM:	0.8942	0.91	0.8802	0.9286
PSNR:	31.01dB	30.25dB	29.76dB	34.60dB
Time:	0.9840s	1.947s	2.8967s	0.134s

Coffee

	AMCD	AFGSA	DEMC	Ours
SSIM:	0.9364	0.8502	0.8293	0.9568
PSNR:	34.14 dB	28.02 dB	25.50 dB	36.04dB
Time:	1.132s	1.265s	3.170s	0.124s

Figure 5. Comparison of results of our approach against the AMCD, DEMC, and AFGSA results.

5. Discussion

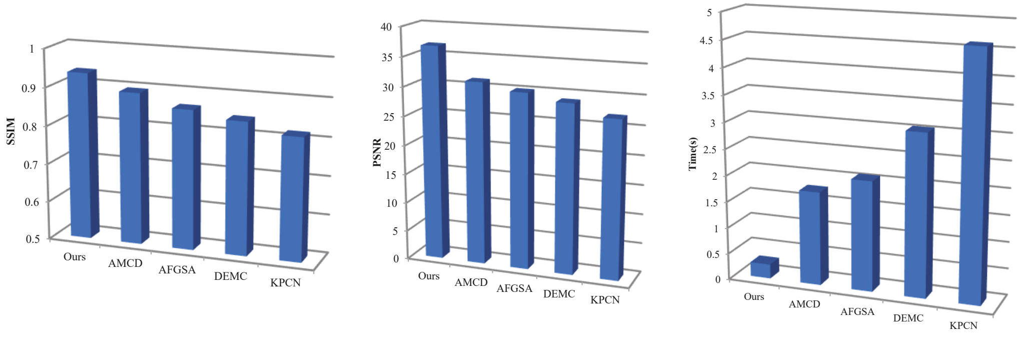
Our main contribution in this approach constitutes a solution for denoising MC renderings trained with a fast deep generative adversarial network, which produces high-quality denoising rendering results with fewer auxiliary buffers, and outperforms state-of-the-art denoising techniques in most situations by saving storage and input/output cost. Furthermore, our approach consistently leads to accurate handling of the diffuse and specular components, in both low-frequency and high-frequency areas, better detail preservation, and a sharp reconstruction to enhance sharp edges with partially saturated pixels and greater detail with less time consumption for rendering. In contrast, the other methods are still time-consuming for denoising in real-time applications even with GPU implementations. The following figure shows the average performance of our work against the baseline of denoising methods KPCN, DEMC, AMCD, and AFGSA.

Figure 6 shows the average performance of our approach against the DEMC, KPCN, AMCD, and AFGSA methods, across test scenes on 4 spp:

In all noise reduction tests, our method is always better than several state-of-the-art solutions. Table 3 shows the aggregate numerical performance of our approach against DEMC, KPCN, AMCD, and AFGSA methods according to the PSNR, SSIM values, and time process for all noise reduction results. Our method consistently has smaller errors, with higher SSIM values and less time consumption than state-of-the-art methods.

Generally, our main contribution in this approach constitutes a solution for denoising MC renderings trained with deep learning, which produces high-quality denoising rendering results with less time-consumption for rendering. In contrast, the other methods are still time-consuming for denoising in real-time applications, even using GPU implementations.

On the other hand, KPCN and DEMC successfully denoise most low-frequency areas. Unfortunately, they fail in high-frequency areas, as only stacking the standard convolution operations makes the network lack resilience when facing different auxiliary features, to make the network restore high-frequency information as much as possible. The AFGSA method loses some details and leads to a wrinkle-like artifact, because it is very aggressive at recovering textures and ignores the specular components. Then, the AMCD method adding the adversarial loss is useful to a certain extent, but they produce smooth results at the junction of high/low-frequency areas due to a smoother global illumination effect. Thus, they cannot essentially eliminate this problem and many other effects. However, in Figure 5 on the floor of the material scene, there are soft shadows on the sharp lines, which cannot be filtered while preserving the sharp edges simultaneously. In contrast, our approach consistently leads to accurate handling of the diffuse and specular components, in both low-frequency and high-frequency areas, and better detail preservation and a sharp reconstruction to enhance sharp edges with partially saturated pixels and greater detail. Moreover, our approach uses fewer auxiliary buffers and outperforms state-of-the-art denoising techniques in most situations by saving storage and input/output cost.



(**a**) (**b**) (**c**)

Figure 6. Average performance and time processes of our approach against DEMC, KPCN, AMCD, and AFGSA. The values are relative to the noisy input (**a**), which shows the performance in the matrix of SSIM, and (**b**) shows the performance in the matrix of PSNR. Accordingly, higher values of SSIM and PSNR mean better performance. Finally, (**c**) shows the comparison of processing time for optimization between our approach and other techniques, whereas the lower values of seconds refer to better performance. Note that the highlighted values mean better performance.

Table 3. Aggregate numerical performance of all methods.

Model	PSNR(dB) ↑	SSIM ↓	Time(s) ↓
KPCN	26.96	0.818	4.612
DEMC	28.63	0.845	3.055
AFGSA	30.07	0.863	2.083
AMCD	31.62	0.895	1.773
Ours	36.76	0.9361	0.2721

6. Conclusions

In this paper, we demonstrated the effect and performance of a kernel prediction network and a deep generative adversarial network to construct an end-to-end general denoising network structure with loss function, and the comparative experiments reflected that our proposed method was effective and had a good denoising effect. In addition, our results were compared with the recent work results of Monte Carlo-rendered image denoising. Accordingly, the comparison results showed that both the visual effects of the image, the measured PSNR, and SSIM showed that our approach had a great improvement against the state-of-the-art. In addition, the denoising effects of the data rendered by multiple renderers showed that the network model of this paper had a relatively good generalization ability and a good adaptability to the rendering data from different rendering systems.

In contrast, to analyze the performance of the method proposed in this paper, we inputted noise images with different sampling rates and compared the denoising effect and running time. The results showed that the method of our approach method achieved better results in terms of effect and running time.

Author Contributions: Conceptualization, A.M.T.A. and C.C.; methodology, A.M.T.A.; software, A.M.T.A.; validation, A.M.T.A.; formal analysis, A.M.T.A.; investigation, A.M.T.A.; resources, C.C.; data curation, A.M.T.A.; writing—original draft preparation, A.M.T.A.; writing—review and editing, A.M.T.A. and C.C.; visualization, A.M.T.A.; supervision, C.C.; project administration, C.C.; funding acquisition, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported partially by the National Natural Science Foundation of China under Grant U19A2063 and partially by the Jilin Provincial Science & Technology Development Program of China under Grant 20190302113GX. The authors would like to thank all reviewers for their valuable comments and suggestions.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MC	Monte Carlo Method
DGAN	Deep Generative Adversarial Network
AFGSA	Auxiliary Feature Guided Self-Attention Module
KPCN	Kernel Predicting Convolutional Network
spp	Samples Per Pixel
SSIM	The Structural Similarity Index
PSNR	Peak Signal-To-Noise Ratio

References

1. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context Encoders: Feature Learning by Inpainting. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; Volume 2, pp. 2536–2544. Available online: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.278 (accessed on 25 November 2021).
2. Brabandere, B.D.; Jia, X.; Tuytelaars, T.; Gool, L.V. Dynamic filter networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 667–675. Available online: https://dl.acm.org/doi/10.5555/3157096.3157171 (accessed on 25 November 2021).
3. Bako, S.; Vogels, T.; McWilliams, B.; Meyer, M.; Novák, J.; Harvill, A.; Sen, P.; Derose, T.; Rousselle, F. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graph.* **2017**, *36*, 1–14. [CrossRef]
4. Vogels, T.; Rousselle, F.; Mcwilliams, B.; Röthlin, G.; Harvill, A.; Adler, D.; Meyer, M.; Novák, J. Denoising with kernel prediction and asymmetric loss functions. *ACM Trans. Graph.* **2018**, *37*, 1–15. [CrossRef]
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
6. Mildenhall, B.; Barron, J.T.; Chen, J.; Sharlet, D.; Ng, R.; Carroll, R. Burst Denoising with Kernel Prediction Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2502–2510. Available online: https://arxiv.org/abs/1712.02327 (accessed on 25 November 2021).
7. Mao, X.-J.; Shen, C.; Yang, Y.-B. Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. *Comput. Vis. Pattern Recognit.* **2016**, *2*, 1–9. Available online: https://arxiv.org/abs/1603.09056 (accessed on 25 November 2021).

8. Gharbi, M.; Li, T.-M.; Aittala, M.; Lehtinen, J.; Durand, F. Sample-based Monte Carlo denoising using a kernel-splatting network. *ACM Trans. Graph.* **2019**, *38*, 125. [CrossRef]
9. Goodfellow, I. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv* **2016**, arXiv:1701.00160. Available online: http://arxiv.org/pdf/1701.00160.pdf (accessed on 25 November 2021).
10. Isola, P.; Zhu, J.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976. [CrossRef]
11. Chen, Q.; Koltun, V. Photographic Image Synthesis with Cascaded Refinement Networks. In Proceedings of the International Conference on Computer Vision (ICCV 2017), Venice, Italy, 22–29 October 2017; pp. 1–10. Available online: https://arxiv.org/abs/1707.09405 (accessed on 26 November 2021).
12. Dosovitskiy, A.; Brox, T. Generating images with perceptual similarity metrics based on deep networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 658–666. [CrossRef]
13. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image Style Transfer Using Convolutional Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423. [CrossRef]
14. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *Comput. Vis. Pattern Recognit.* **2016**, *9906*, 694–711. Available online: https://arxiv.org/abs/1603.08155 (accessed on 15 November 2021).
15. Bui, G.; Le, T.; Morago, B.; Duan, Y. Point-based rendering enhancement via deep learning. *Vis. Comput.* **2018**, *34*, 829–841. [CrossRef]
16. Xu, B.; Zhang, J.; Wang, R.; Xu, K.; Yang, Y.-L.; Li, C.; Tang, R. Adversarial Monte Carlo Denoising with Conditioned Auxiliary Feature Modulation. *ACM Trans. Graph.* **2019**, *38*, 1–12. [CrossRef]
17. Yang, X.; Wang, D.; Hu, W.; Zhao, L.-J.; Yin, B.-C.; Zhang, Q.; Wei, X.-P.; Fu, H. DEMC: A Deep Dual-Encoder Network for Denoising Monte Carlo Rendering. *J. Comput. Sci. Technol.* **2019**, *34*, 1123–1135. [CrossRef]
18. Munkberg, J.; Hasselgren, J. Neural Denoising with Layer Embeddings. *Comput. Graph. Forum* **2020**, *39*, 1–12. [CrossRef]
19. Lu, Y.; Xie, N.; Shen, H.T. DMCR-GAN: Adversarial Denoising for Monte Carlo Renderings with Residual Attention Networks and Hierarchical Features Modulation of Auxiliary Buffers. In Proceedings of the SIGGRAPH Asia 2020 Technical Communications, Virtual Event, Korea, 1–9 December 2020; pp. 1–4. [CrossRef]
20. Yu, J.; Nie, Y.; Long, C.; Xu, W.; Zhang, Q.; Li, G. Monte Carlo denoising via auxiliary feature guided self-attention. *ACM Trans. Graph.* **2021**, *40*, 1–13. [CrossRef]
21. Marinč, T.; Srinivasan, V.; Gül, S.; Hellge, C.; Samek, W. Multi-Kernel Prediction Networks for Denoising of Burst Images. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 2404–2408. [CrossRef]
22. Chaitanya, C.R.A.; Kaplanyan, A.S.; Schied, C.; Salvi, M.; Lefohn, A.; Nowrouzezahrai, D.; Aila, T. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graph.* **2017**, *36*, 1–12. [CrossRef]
23. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680. Available online: https://dl.acm.org/doi/10.5555/2969033.2969125 (accessed on 26 December 2021).
24. Lee, W.-H.; Ozger, M.; Challita, U.; Sung, K.W. Noise Learning Based Denoising Autoencoder. *IEEE Commun. Lett.* **2021**, *25*, 2983–2987. [CrossRef]
25. Fan, L.; Zhang, F.; Fan, H.; Zhang, C. Brief review of image denoising techniques. *Vis. Comput. Ind. Biomed. Art* **2019**, *2*, 7. [CrossRef]
26. Horé, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369. [CrossRef]
27. Reinhard, E.; Stark, M.; Shirley, P.; Ferwerda, J. Photographic tone reproduction for digital images. *ACM Trans. Graph.* **2002**, *21*, 267–276. [CrossRef]
28. Bitterli, B. Rendering Resources. 2016, vol. 9. Available online: https://benedikt-bitterli.me/resources/ (accessed on 1 November 2021).
29. Kingma, D.P.; Welling, M. An Introduction to Variational Autoencoders. *Found. Trends Mach. Learn.* **2019**, *12*, 307–392. Available online: https://arxiv.org/abs/1906.02691 (accessed on 26 November 2021). [CrossRef]
30. Zhou, W.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
31. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256. Available online: http://proceedings.mlr.press/v9/glorot10a.html (accessed on 27 November 2021).
32. Bitterli, B. The Tungsten Renderer. 2014. Available online: https://benedikt-bitterli.me/tungsten.html (accessed on 1 November 2021).

symmetry

MDPI

Article

Estimation of Symmetry in the Recognition System with Adaptive Application of Filters

Volodymyr Hrytsyk [1], Mykola Medykovskyy [1] and Mariia Nazarkevych [2,*]

[1] Department of Automated Control Systems, Lviv Polytechnic National University, 79013 Lviv, Ukraine; volodymyr.v.hrytsyk@lpnu.ua (V.H.); mykola.o.medykovskyi@lpnu.ua (M.M.)
[2] Department of Information Technology Publishing, Lviv Polytechnic National University, 79013 Lviv, Ukraine
* Correspondence: mariia.a.nazarkevych@lpnu.ua

Abstract: The aim of this work is to study the influence of lighting on different types of filters in order to create adaptive systems of perception in the visible spectrum. This problem is solved by estimating symmetry operations (operations responsible for image/image transformations). Namely, the authors are interested in an objective assessment of the possibility of reproducing the image of the object (objective symmetry of filters) after the application of filters. This paper investigates and shows the results of the most common edge detection filters depending on the light level; that is, the behavior of the system in a room with indirect natural and standard (according to the requirements of the educational process in Ukraine) electric lighting was studied. The methods of Sobel, Sobel x, Sobel y, Prewitt, Prewitt x, Prewitt y, and Canny were used and compared in experiments. The conclusions provide a subjective assessment of the performance of each of the filters in certain conditions. Dependencies are defined that allow giving priority to certain filters (from those studied) depending on the lighting.

Keywords: edge detectors; PSNR evaluation; adaptive system

Citation: Hrytsyk, V.; Medykovskyy, M.; Nazarkevych, M. Estimation of Symmetry in the Recognition System with Adaptive Application of Filters. *Symmetry* **2022**, *14*, 903. https://doi.org/10.3390/sym14050903

Academic Editors: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Received: 16 February 2022
Accepted: 28 March 2022
Published: 28 April 2022

1. Introduction

In the last decade, the possibilities of autonomous decision making by computer vision systems have been actively studied. The topic of autonomous agents to support solutions that are able to adapt to the environment is relevant in a variety of applications: medicine [1–3], systems and means of artificial intelligence [4–6]. The problem includes the question of where fuzzy logic is needed [7], security issues [8–11] and biometric recognition systems [12,13], systems to support people with disabilities and related technologies and applications aimed to include wayfinding and navigation [5,13], land research, space research, agricultural issues, industrial climate issues, smart things, etc.

The task of automated selection of key characteristics for the classification of images using computer tools is not a trivial problem [14–16]; especially for the variable field of attention [17,18]. There are many methods and algorithms for identifying key characteristics for image classification [19–24], but each has its disadvantages and advantages. Most of the existing methods that solve this problem are effective only for individual objects, such as human faces, simple geometric shapes, and handwritten or printed symbols, but only under certain conditions, including certain lighting and the position of the object from the wearable camera and background. An urgent problem now is the creation of automated systems that compensate for the limited capabilities of people at different levels. When using artificial neural networks [19,20], reducing the amount of computation in learning is an important element to teach mathematical options for processing. This paper considers the problem of comparing the methods of selection of characteristic features under different external conditions in order to identify the best method for given conditions. Adaptive algorithms for detecting, classifying, and tracking the edges of objects have been developed in [25]. Adaptation is achieved by using thresholds to obtain contours. The creation of contours is achieved through the Kalman filter, the use of which significantly improves the computation time, as well as provides the system with the required accuracy. In [26], invariant approaches to structural features were used to find facial features, texture, shape, and color of the skin regardless of changes in lighting. Statistical models have been developed, which are the basis for testing the model. The advantage of the proposed method is that it can detect faces of different sizes and different poses without restrictions on lighting conditions. However, the spectral characteristics of the face / skin are not studied in this work. In [27], for an image that is under background and lighting conditions, a method for determining the edges based on the advanced arithmetic operator Prewitt was proposed. An improved preprocessing operation was performed in the work. The characteristics of horizontal projection and vertical location of the upper and lower edges for positioning were used. The results of the experiments show that the algorithm has high speed, positioning speed, and good practical value.

Table 1 shows the increase in absolute contrast values from the state of the object. The state of the object can change tens of thousands of times, and up to one million times when exposed to the sun. Because glare affects the brightness of an object, the brightness of the sun's disk can reach up to 108 lux. Shade illumination can be reduced to 100 lux. Today, it is impossible to observe several objects with tens of thousands of different illuminations on the same camera. Therefore, the inevitable loss of information.

Table 1. Lighting on the object in the Suites.

Intensity Scale	Lighting on the Object in the Suites	Inside the Near Window, Lux/Time
10^8	The Sun disc is at noon	
10^6	Glitter of water and metal under sunlight	
10^4	Snow-covered snow and clouds. Illuminated objects by day	2300/13:30 260/13:00
10^2	Objects in the shade, in the afternoon	100/16:50 70/16:00 (the sun came out of the clouds) 50/16:30
10^0	Moon	
10^{-2}	Stars	

The aim of the work is to create adaptive systems of perception in the visible spectrum by constructing the dependences of the quality of the applied filters in dynamically changing conditions. In particular, the paper studies the behavior (in terms of symmetry of object representation) of the most popular methods of detecting edges under different lighting conditions.

2. Materials and Methods

2.1. Sobel's Operator

The Sobel operator is a discrete differential operator that calculates the approximate value of the image gradient [28]. The result of the application of the Sobel operator at each point of the image is either the brightness gradient vector at this point or its norms.

2.1.1. Description

The image convolution on which the Sobel operator is based is performed by small separable integer filters in the vertical and horizontal directions. The Sobel operator uses a gradient approximation that is not accurate, and this becomes especially noticeable at high-frequency image oscillations.

At each point of the image, the brightness gradient is calculated by the Sobel operator. Thus, there is the direction of the greatest increase in brightness and change of this value. Changing the brightness indicates the smoothness or sharpness of the change at each point of the image, the probability of finding a point on the border, and hence the orientation of the contour. This calculation is more reliable and simpler than calculating the direction of orientation.

A two-dimensional vector is a gradient of the function of two variables for each point of the image. Its components are derivatives of image brightness, which are calculated horizontally and vertically. The result of the work of the Sobel operator will be a zero vector at the point of the region of constant brightness and at the point lying on the boundary of the regions, a vector with the direction of increasing brightness.

2.1.2. Formulation

Strictly speaking, the operator uses 3 × 3 cores to calculate output values. If A is the original image, and G_x and G_y are two images with approximate points of derivatives in x and y:

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A, \text{ and } G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \tag{1}$$

where $*$ means a two-dimensional convolution operation.

The x-coordinate here increases "to the right", and y "down". At each point of the image, the approximate value of the gradient value can be calculated using the obtained approximate values of the derivatives (meaning element by element):

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{2}$$

Using this information, we can also calculate the direction of the gradient:

$$\Theta = arctan\left(\frac{G_x}{G_y}\right) \tag{3}$$

where, for example, the angle Θ is zero for the vertical boundary in which the dark side is on the left.

The brightness function is known to us at discrete points, we need to determine the differentiated function that passes through these points. Derivatives at any single point are functions of brightness from all points of the image. The solution of the derivatives can be calculated with a certain degree of accuracy.

Sobel's operator is an inaccurate approximation of the image gradient, but it is high enough for practical application in many problems. Specifically, the operator uses the intensity value only around 3 × 3 of each pixel to obtain an approximation of the corresponding image gradient and uses only integer values of the luminance weights to estimate the gradient.

2.1.3. Extension to Another Number of Dimensions

The Sobel operator consists of two separate operations:

- Smoothing with a triangular filter perpendicular to the derivative direction:

$$h(-1) = 1,\ h(0) = 2,\ h(1) = 1$$

- Finding a simple central change in the direction of the derivative:

$$h(-1) = 1,\ h'^{(0)} = 0,\ h'^{(1)} = -1$$

Sobel filters for image derivatives in various dimensions for

$$\begin{gathered} y, z, t \in (0, -1, 1) \\ 1\text{D}:\ h'_x(x) = h\prime(x) \\ 2\text{D}:\ h'_x(x, y) = h\prime(x)h(y) \\ 3\text{D}:\ h'_x(x, y, z) = h\prime(x)h(y)h(z) \\ 4\text{D}:\ h'_x(x, y, z, t) = h\prime(x)h(y)h(z)h(t) \end{gathered}$$

Here is an example of a three-dimensional Sobel core for an axis z:

$$h\prime_z(:, :, -1) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$h\prime_z(:, :, 0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$h\prime_z(:, :, 1) = \begin{bmatrix} -1 & -2 & -1 \\ -2 & -4 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$

2.1.4. Technical Details

As follows from the definition, the Sobel operator can be implemented by simple hardware and software. Approximating the vector gradient requires only eight pixels around point x of the image and integer arithmetic. Moreover, both discrete filters described above can be separated:

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

and two derivatives, G_x and G_y, can now be calculated as

$$G_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \cdot A;\ G_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \cdot \left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \cdot A \right)$$

The resolution of these calculations can reduce the arithmetic operations with each pixel.

2.2. Canny's Operator

Canny's operator (Canny boundary detector, Canny algorithm) is a computer vision detection operator in the field of computer vision. It was developed in 1986 by John F. Canny and uses a multistep algorithm to detect a wide range of boundaries in images [16,29]. Canny said that this filter could be well approximated by the first Gaussian derivative. Kenny introduced the concept of non-maximum suppression, which means that pixels of boundaries are declared pixels in which the local maximum of the gradient is reached in the direction of the gradient vector.

- Good detection (Kenny interpreted this property as increasing the signal-to-noise ratio);

- Good localization (correct determination of the edge position);
- A single response to one edge.

From these criteria, the target function of error cost was then built, the minimization of which is the "optimal" linear operator for convolution with images.

The boundary detector algorithm is not limited to calculating the gradient of the smoothed image. Only the maximum points of the image gradient remain in the border contour, and the maximum points lying near the border are removed. It also uses information about the direction of the border in order to remove the points right next to the border and not to break the border itself near the local gradient maxima. Then, with the help of two thresholds, weak borders are removed. The fragment of the border is processed as a whole. If the value of the gradient somewhere on the observed fragment exceeds the upper threshold; this fragment also remains on the "permissible" edge and in places where the value of the gradient falls below this threshold, until it falls below the lower threshold. If there is no point on the whole fragment with a value greater than the upper threshold, it is deleted. This hysteresis reduces the number of breaks in the original edges. The inclusion of noise attenuation in Kenny's algorithm increases the stability of the results on the one hand and, on the other hand, increases computational costs and leads to distortion and even loss of boundary details. For example, this algorithm rounds the corners of objects and destroys boundaries at connection points.

The main stages of Canny's algorithm are:

Smoothing: blurred images to remove noise. Canny's operator uses a filter that may be close to the first Gaussian derivative σ = 1.4:

$$B = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \cdot A$$

Search for gradients: borders are marked where the image gradient becomes maximum. They can have different directions, so Canny's algorithm uses four filters to detect horizontal, vertical, and diagonal edges in a blurred image. At this stage, Formulas (2) and (3) are used.

The angle of the direction of the gradient vector [30] is rounded and can take the following values: 0, 45, 90, and 135 (see Figure 1).

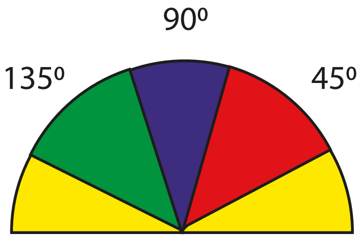


Figure 1. Gradient direction.

Suppression of non-maxima.

Only local maxima are marked as edges.

Double threshold filtering: potential limits are determined by thresholds.

Trace the area of ambiguity. The final limits are determined by suppressing all edges that are not related to certain (strong) limits.

Before using the detector, the image is usually converted to grayscale to reduce computational costs. This stage is typical of many image-processing methods.

2.3. Prewitt's Operator

Prewitt operator is a method of selecting boundaries in image processing, which calculates the maximum response on the set of convolution cores to find the local orientation of the border in each pixel. It was created by Dr. Judith Prewitt to identify the boundaries of medical imaging [31].

Different kernels are used for this operation. From one core, you can obtain eight, rearranging the coefficients in a circle. Each result will be sensitive to the direction of the limit from 0 to 315 with a step of 45, where 0 corresponds to the vertical limit. The maximum response of each pixel is the value of the corresponding pixel in the original image. Its values are between 1 and 8, depending on the number of nuclei that provide the greatest result.

This method of edge detection is also called edge template matching because the image is mapped to a set of templates, and each represents some boundary orientation. The size and orientation of the border in a pixel is then determined by the pattern that best matches the local neighborhood of the pixel.

While a differential gradient detector requires a time-consuming calculation of the orientation estimate for magnitudes in the vertical and horizontal directions, the Prewitt limit detector provides a direct direction from the nucleus with maximum result. The set of nuclei is limited to 8 possible directions, but experience shows that most direct estimates of orientation are also not very accurate. On the other hand, a set of cores requires 8 convolutions for each pixel, while a set of gradients of the gradient method requires only 2: sensitive vertically and horizontally.

Formulation

The operator uses two 3 × 3 cores, collapsing the original image to calculate the approximate values of the derivatives: one horizontally and one vertically. Let A be the original image and G_x i G_y be two images in which each point contains a horizontal and vertical approximation of the derivative, which is calculated as:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \cdot A,\ and\ G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \cdot A$$

3. Experiments

3.1. Experiment №1. 16:00, Action at 67–70 lux

The program on the smartphone was used to measure the suites, as the study was performed to link to the most common video cameras. Figures 2–29 shows the work of the considered methods in comparison with the original image. Images are grouped according to lighting conditions. Thus, the first experiment (lighting 70 lux) is presented in Figure 2 (original input image and the result of processing by the Sobel method); Figure 3 (normalized Sobel gradient on the X axis and gradient on the Y axis); Figure 4 (Canny and Pruitt method); Figure 5 (Prewitt's operator on the X axis and (Prewitt's operator on the Y axis).

(**a**) (**b**)

Figure 2. (**a**) Input image; (**b**) Sobel gradient at 70 lux.

(**a**) (**b**)

Figure 3. (**a**) Normalized Sobel gradient of the image along the *x*-axis; (**b**) normalized Sobel gradient of the image on the *y*-axis at 70 lux.

(**a**) (**b**)

Figure 4. (**a**) Canny's operator; (**b**) Prewitt's operator at 70 lux.

(**a**) (**b**)

Figure 5. (**a**) Prewitt's operator X; (**b**) Prewitt's operator Y at 70 lux.

(**a**) (**b**)

Figure 6. (**a**) Input image; (**b**) Sobel operator at 50 lux.

(**a**) (**b**)

Figure 7. (**a**) Sobel operator on the *x*-axis; (**b**) Sobel operator on the *y*-axis at 50 lux.

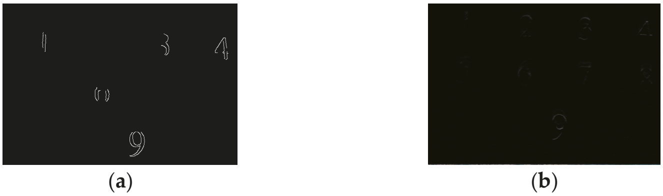

(**a**) (**b**)

Figure 8. (**a**) Canny operator; (**b**) Prewitt's operator on the *y*-axis at 50 lux.

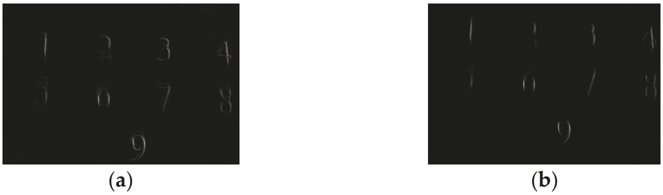

(**a**) (**b**)

Figure 9. (**a**) Prewitt's operator on the *x*-axis; (**b**) Prewitt's operator on the *y*-axis at 50 lux.

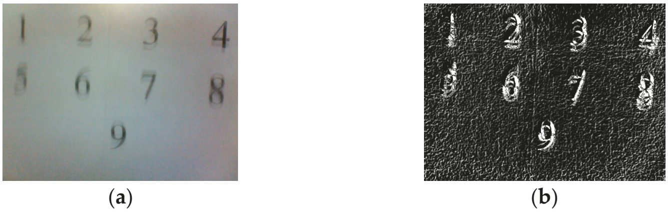

(**a**) (**b**)

Figure 10. (**a**) Input image; (**b**) Sobel operator at 100 lux.

(**a**) (**b**)

Figure 11. (**a**) Sobel operator on the *x*-axis; (**b**) Sobel operator on the *y*-axis at 100 lux.

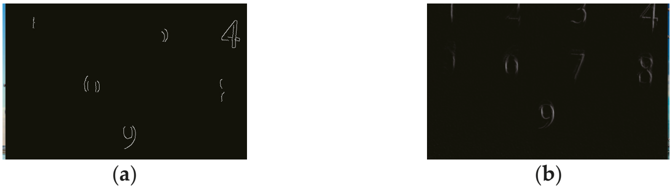

(**a**) (**b**)

Figure 12. (**a**) Canny operator; (**b**) Prewitt's operator on the *y*-axis at 100 lux.

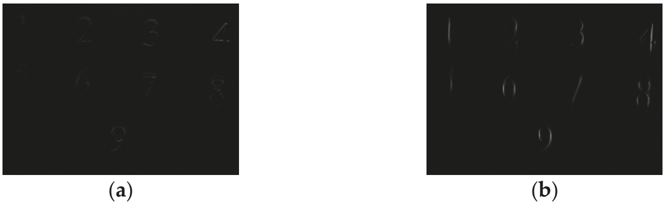

(**a**) (**b**)

Figure 13. (**a**) Prewitt's operator on the *x*-axis; (**b**) Prewitt's operator on the *y*-axis at 100 lux.

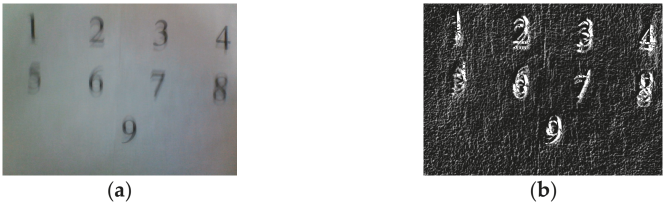

(**a**) (**b**)

Figure 14. (**a**) Input image; (**b**) Sobel operator at 260 lux.

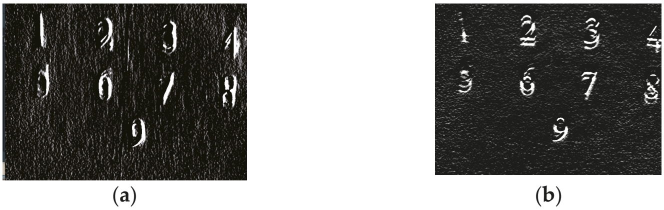

(**a**) (**b**)

Figure 15. (**a**) Sobel operator on the *x*-axis; (**b**) Sobel operator on the *y*-axis at 260 lux.

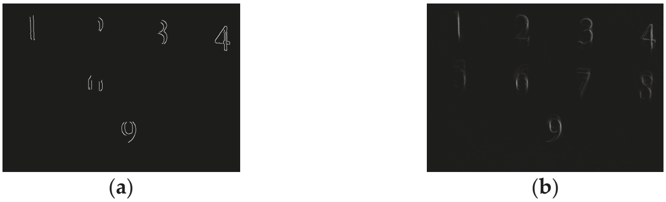

(**a**) (**b**)

Figure 16. (**a**) Canny operator; (**b**) Prewitt's operator at 260 lux.

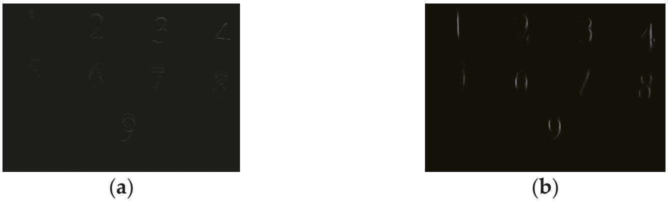

(**a**) (**b**)

Figure 17. (**a**) Prewitt's operator X; (**b**) Prewitt's operator Y at 260 lux.

Figure 18. (**a**) Input image; (**b**) Sobel operator at 2300 lux.

Figure 19. (**a**) Sobel operator on the *x*-axis; (**b**) Sobel operator on the *y*-axis at 2300 lux.

Figure 20. (**a**) Canny operator; (**b**) Prewitt's operator at 2300 lux.

Figure 21. (**a**) Prewitt's operator X; (**b**) Prewitt's operator Y at 2300 lux.

Figure 22. (**a**) input image; (**b**) Sobel operator at 2300 lux+ man shadow.

Figure 23. (**a**) Sobel operator on the *x*-axis; (**b**) Sobel operator on the *y*-axis at 2300 lux+ man shadow.

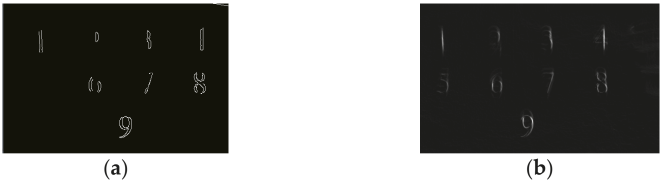

(**a**) (**b**)

Figure 24. (**a**) Canny operator; (**b**) Prewitt's operator at 2300 lux+ man shadow.

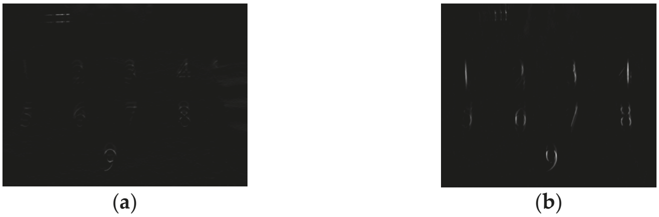

(**a**) (**b**)

Figure 25. (**a**) Prewitt's operator X; (**b**) Prewitt's operator y at 2300 lux+ man shadow.

(**a**) (**b**)

Figure 26. (**a**) input image; (**b**) Sobel operator at 2350 lux.

(**a**) (**b**)

Figure 27. (**a**) Sobel operator on the *x*-axis; (**b**) Sobel operator on the *y*-axis at 2350 lux.

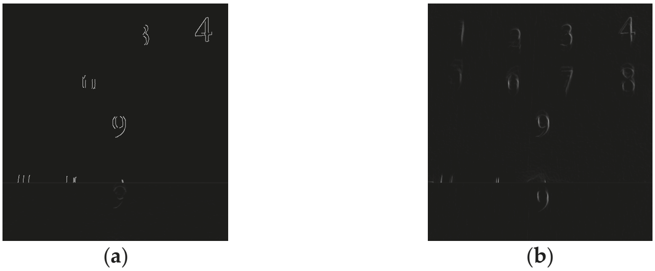

(**a**) (**b**)

Figure 28. (**a**) Canny operator; (**b**) Prewitt's operator at 2350 lux.

(**a**) (**b**)

Figure 29. (**a**) Prewitt's operator X; (**b**) Prewitt's operator Y at 2350 lux.

3.2. Experiment №2. 16:30, Action at 50 Lux

The second experiment (lighting 50 lux) is presented in Figure 6 (original input image and the result of processing by the Sobel method); Figure 7 (normalized Sobel gradient on the X-axis and gradient on the Y-axis); Figure 8 (Canny's and Prewitt's method); Figure 9 (Prewitt's operator on the X-axis and (Prewitt's operator on the Y-axis).

3.3. Experiment №3. 16:50, Action at 100 Lux (Includes E-Electric Lighting)

The third experiment (lighting 100 lux) is presented in Figure 10 (original input image and the result of processing by the Sobel method); Figure 11 (normalized Sobel gradient on the X-axis and gradient on the Y-axis); Figure 12 (Canny's and Prewitt's method); Figure 13 (Prewitt's operator on the X-axis and (Prewitt's operator on the Y-axis).

3.4. Experiment №3. 13:00, Action at 260 Lux

The forth experiment (lighting 260 lux) is presented in Figure 14 (original input image and the result of processing by the Sobel method); Figure 15 (normalized Sobel gradient on the X-axis and gradient on the Y-axis); Figure 16 (Canny's and Prewitt's method); Figure 17 (Prewitt's operator on the X-axis and (Prewitt's operator on the Y-axis).

3.5. Experiment №3. 13:30, Action at 2300 Lux

The fifth experiment (lighting 2300 lux) is presented in Figure 18 (original input image and the result of processing by the Sobel method); Figure 19 (normalized Sobel gradient on the X-axis and gradient on the Y-axis); Figure 20 (Canny's and Prewitt's method); Figure 21 (Prewitt's operator on the X-axis and (Prewitt's operator on the Y-axis).

3.6. Experiment №3. 13:30, Action at 2300 Lux + Man Shadow

The sixth experiment (lighting 2300 lux) is presented in Figure 22 (original input image and the result of processing by the Sobel method); Figure 23 (normalized Sobel gradient on the X-axis and gradient on the Y-axis); Figure 24 (Canny's and Prewitt's method); Figure 25 (Prewitt's operator on the X-axis and (Prewitt's operator on the Y-axis).

3.7. Experiment №3. 13:30, Action at 2350 Lux, Cam Front to Light

The seventh experiment (lighting 2350 lux) is presented in Figure 26 (original input image and the result of processing by the Sobel method); Figure 27 (normalized Sobel gradient on the X-axis and gradient on the Y-axis); Figure 28 (Canny's and Prewitt's method); Figure 29 (Prewitt's operator on the X-axis and (Prewitt's operator on the Y-axis).

4. Results

Figure 30 shows the dependences of the objective comparison of distances (PSNR) between the original value of the brightness of the pixel with the values obtained after applying the appropriate filters. The data in the chart are summarized.

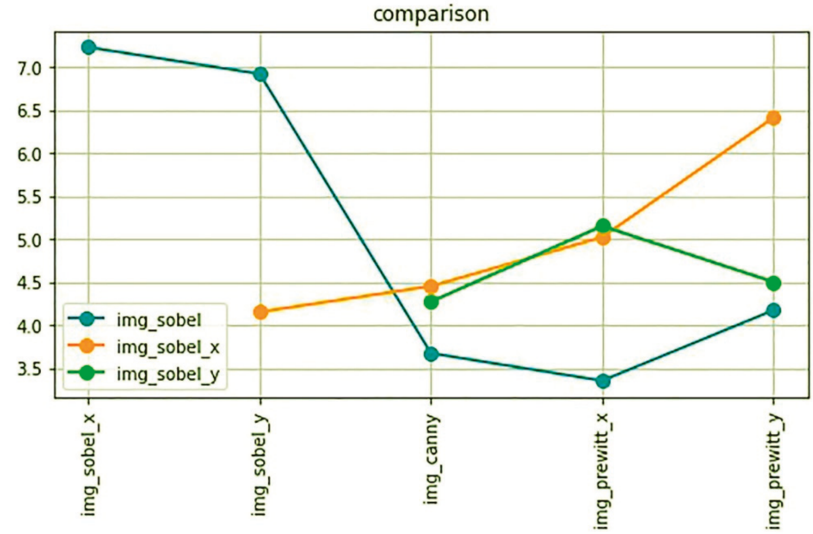


Figure 30. Results of objective assessment (PSNR) of symmetry of researched operations.

The experiment was performed as follows. The original image was illuminated at 50 lux, 70 lux, 100, 260, and 2300 lux. These images are presented in Figures 2, 6, 10, 14, 18, 22 and 26, respectively. Canny, Sobel, and Prewitt filters were used for these images. These filtered images were compared with each other to assess the best performance of the filter and to make recommendations for the use of certain filters. Diagram 1 shows three graphs of the relationship between the signal-to-noise ratio of the filtered images. The blue color shows the values of the signal-to-noise ratio between img_sobel and img_sobel_x. This value takes the value of 7.23 and is the largest value in this chart, and therefore, they are the most similar to each other. The second value is the result of comparing img_sobel and img_sobel_y and takes the value of 6.92. We also observe high PSNR values compared to others, which explains that very similar transformations were used. The other values are a comparison between the pairs img_sobel and img_cany and img_sobel and img_prewitt_x, and img_sobel and img_prewitt_y. Because filtering was applied by other filters, the images are less similar, and that makes sense. Charts of other colors, similar to img_sobel_x and img_sobel_y, work similarly. Slightly higher values between the pairs img_sobel_x and img_prewitt_y are explained by the better finding of contours in the image, which should be taken into account when choosing one filtering method.

We performed an experiment in which the edges of the source objects were marked on the input image in one of the raster graphics editors. Figure 31a shows the image taken at 50 lux, Figure 31b shows the filtered image of the Sobel operator, Figure 31c shows the Canny operator, and Figure 31d shows Prewitt's operator. As a result of filtering, more edges from the original image were formed in Prewitt and Canny, which is seen subjectively. An experiment was performed to compare the input image with the filtered images using PSNR, and the results are shown:

Input image, img_sobel—11.7;
Input image, img_canny—13.76;
Input image, img_prewitt's—14.67.

This is best performed by Prewitt's operator, as the PSNR values are the highest.

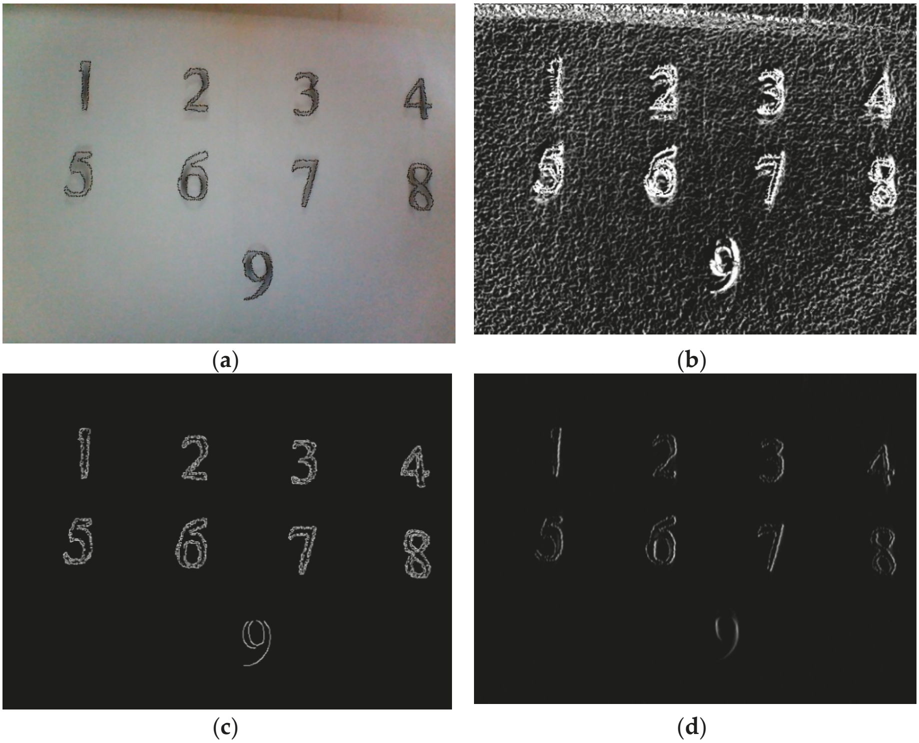


Figure 31. (**a**) Input image; (**b**) Sobel operator; (**c**) Canny operator; (**d**) Prewitt's operator.

5. Discussion

The experiments show that to create an adaptive system, you can use the technique of selecting filters, for which you need to choose a filter that provides the best symmetry of the images under appropriate conditions. The success of the experiments inspires the authors to further test all the most widely used filters in different conditions, which will allow (according to an objective assessment) choosing filters that provide the best symmetry of image display for specific conditions.

In image segmentation evaluation, the structural similarity index (SSIM) estimates the visual impact of shifts in an image [26]. The SSIM consists of three local comparison functions, namely luminance comparison, contrast comparison, and structure comparison, between two signals excluding other remaining errors. The SSIM is computed locally by moving an 8 × 8 window for each pixel, unlike the peak signal-to-noise ratio (PSNR) or root-mean-square error (RMSE), which are measured at the global level. "Even though SSIM can be applied in the case of an edge detection evaluation, in the presence of too many areas without contours, the obtained score is not efficient or useful (in order to judge the quality of edge detection with the SSIM, it is necessary to compare with an image having detected edges situated throughout the image areas)" [26]. This is why PSNR was chosen as a simple and widespread method of global assessment. Other popular evaluation methods [26] do not have strong superiority and have less popularity. Due to this, this paper had an unsettled task of studying the behavior of filters in a particular education (without strong differences).

The success of this study encourages the authors to expand the number of methods and conditions in the next study, in particular to conduct experimental studies of additional methods [27]. This paper analyzes the speed in contrast to the idea of lighting level.

6. Conclusions

The conducted experiments show that among the considered methods, in all conditions, only variations of the Sobel operator compete for the title of the best. With different lighting, different variations can be considered the best. In addition, although there is a difference in quality between variations of the Sobel operator, with all types of lighting, they all give a good result. All others highlight the characteristics rather weakly. The resulting dependence presented in the diagram allows fully automating the process of selecting a filter for image preprocessing when designing adaptive computer vision.

Author Contributions: Conceptualization, V.H. and M.M.; methodology, M.M.; software, M.N.; validation, M.N., M.N.; formal analysis, V.H.; investigation, M.M.; resources, M.N.; data curation, M.N.;writing—original draft preparation, V.H.; writing—review and editing, M.N.; visualization, M.N.; supervision, V.H.; project administration, V.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Boyko, I.; Petryk, M.; Fraissard, J. Investigation of the electron-acoustic phonon interaction via the deformation and piezoelectric potentials in AlN/GaN resonant tunneling nanostructures. *Superlattices Microstruct.* **2021**, *156*, 106928. [CrossRef]
2. Petryk, M.R.; Boyko, I.V.; Khimich, O.M.; Petryk, M.M. High-Performance Supercomputer Technologies of Simulation and Identification of Nanoporous Systems with Feedback for n-Component Competitive Adsorption. *Cybern. Syst. Anal.* **2021**, *57*, 316–328. [CrossRef]
3. Boyko, I.; Mudryk, I.; Petryk, M.; Petryk, M. High-Performance Adsorption Modeling Methods with Feedback-Influynces in n-component Nanoporous Media. In Proceedings of the 2021 11th International Conference on Advanced Computer Information Technologies, Deggendorf, Germany, 15–17 September 2021; pp. 441–444.
4. Mashkov, O.; Krak, Y.; Babichev, S.; Bardachov, Y.; Lytvynenko, V. *Preface Lecture Notes on Data Engineering and Communications*; Springer: Berlin, Germany, 2022; Volume 77, pp. v–vi.
5. Hrytsyk, V.; Nazarkevych, M. Real-Time Sensing, Reasoning and Adaptation for Computer Vision Systems. In *ISDMCI 2021: Lecture Notes in Computational Intelligence and Decision Making*; Springer: Cham, Switzerland, 2021; pp. 573–585.
6. Hrytsyk, V.; Grondzal, A.; Bilenkyj, A. Augmented reality for people with disabilities. In Proceedings of the 2015 Xth International Scientific and Technical Conference" Computer Sciences and Information Technologies" (CSIT), Lviv, Ukraine, 14–17 September 2015; pp. 188–191.
7. Beucher, A.; Fröjdö, S.; Österholm, P.; Martinkauppi, A.; Edén, P. Fuzzy Logic for Acid Sulfate SoilMapping: Application to the Southern Part of the Finnish Coastal Areas. *Geoderma* **2014**, *226–227*, 21–30. [CrossRef]
8. Nazarkevych, M.; Hrytsyk, V.; Voznyi, Y.; Marchuk, A.; Vozna, O. Method of detecting special points on biometric images based on new filtering methods. *Proc. CEUR Workshop Proc.* **2021**, 243–251. Available online: http://ceur-ws.org/Vol-2923/paper26.pdf (accessed on 26 March 2022).
9. Nazarkevych, M.; Dmytruk, S.; Hrytsyk, V.; Maslanych, I.; Sheketa, V. Evaluation of the effectiveness of different image skeletonization methods in biometric security systems. *Int. J. Sens. Wirel. Commun. Control* **2021**, *11*, 542–552. [CrossRef]
10. Kamińska, D.; Aktas, K.; Rizhinashvili, D.; Moeslund, T.B.; Anbarjafari, G. Two-stage recognition and beyond for compound facial emotion recognition. *Electronics* **2021**, *10*, 2847. [CrossRef]
11. Aakerberg, A.; Nasrollahi, K.; Moeslund, T.B. Real-world super-resolution of face-images from surveillance cameras. *IET Image Processingthis* **2022**, *16*, 442–452. [CrossRef]
12. Nazarkevych, M.; Voznyi, Y.; Hrytsyk, V.; Havrysh, B.; Lotoshynska, N. Identification of biometric images by machine learning. In Proceedings of the 2021 IEEE 12th International Conference on Electronics and Information Technologies (ELIT), Lviv, Ukraine, 19–21 May 2021; pp. 95–98.
13. Nazarkevych, M.; Lotoshynska, N.; Hrytsyk, V.; Vozna, O.; Palamarchuk, O. Design of biometric system and modeling of machine learning for entering the information system. *Int. Sci. Tech. Conf. Comput. Sci. Inf. Technol.* **2021**, *2*, 225–230.
14. Prandi, C.; Barricelli, B.R.; Mirri, S.; Fogl, D. Accessible wayfinding and navigation: A systematic mapping study. *Univers. Access Inf. Soc.* **2021**, 1–28. [CrossRef]
15. Duda, O.R.; Hart, E.P. *Pattern Classification and Scene Analysis*; Wiley: New York, NY, USA, 1973; 482p.

16. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In Proceedings of the Eighth IEEE International Conference on Computer Vision, ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001; Volume 23.
17. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 679–698. [CrossRef]
18. Lindeberg, T. Normative theory of visual receptive fields. *Heliyon* **2021**, *7*, e05897. [CrossRef] [PubMed]
19. Jansson, Y.; Lindeberg, T. Dynamic Texture Recognition Using Time-Causal and Time-Recursive Spatio-Temporal Receptive Fields. *J. Math. Imaging Vis.* **2018**, *60*, 1369–1398. [CrossRef]
20. Beucher, A.; Rasmussen, C.B.; Moeslund, T.B.; Greve, M.H. Interpretation of Convolutional Neural Networks for Acid Sulfate Soil Classification. *Front. Environ. Sci.* **2022**, *9*, 809995. [CrossRef]
21. Rasmussen, C.B.; Lejbølle, A.R.; Nasrollahi, K.; Moeslund, T.B. *Evaluation of Edge Platforms for Deep Learning in Computer Vision*; Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics; LNCS: Berlin, Germany, 2021; Volume 12664, pp. 523–537.
22. Alarcao, S.M.; Fonseca, M.J. Emotions recognition using EEG signals: A survey. *IEEE Trans. Affect. Comput.* **2017**, *10*, 374–393. [CrossRef]
23. Mamun, I. Image Classification Using SSIM. Towards Data Science. Available online: https://towardsdatascience.com/image-classification-using-ssim-34e549ec6e12 (accessed on 26 March 2022).
24. Everts, I.; Van Gemert, J.C.; Gevers, T. Evaluation of color spatio-temporal interest points for human action recognition. *IEEE Trans. Image Processing* **2014**, *23*, 1569–1580. [CrossRef]
25. Spontón, H.; Cardelino, J. A Review of Classic Edge Detectors. *Image Processing Line* **2015**, *5*, 90–123. [CrossRef]
26. Magnier, B. Edge detection: A review of dissimilarity evaluations and a proposed normalized measure: Review. *Multimed. Tools Appl.* **2018**, *77*, 9489–9533. [CrossRef]
27. Tariq, N.; Hamzah, R.A.; Ng, T.F.; Wang, S.L.; Ibrahim, H. Quality Assessment Methods to Evaluate the Performance of Edge Detection Algorithms for Digital Image: A Systematic Literature Review. *IEEE Access* **2021**, *9*, 87763–87776. [CrossRef]
28. Sobel, I.; Feldman, G. A 3 × 3 Isotropic Gradient Operator for Image Processing. *Pattern Classif. Scene Anal.* **1973**, 271–272.
29. Canny, J.F. Finding Edges and Lines in Images. Master's Thesis, MIT AI Laboratory, Cambridge, CA, USA, 1983; 720p.
30. Ding, L.; Goshtasby, A. On the Canny edge detector. *Pattern Recognit.* **2001**, *34*, 721–725. [CrossRef]
31. Prewitt, J.M. Object enhancement and extraction. *Pict. Processing Psychopictorics* **1970**, *10*, 15–19.

applied sciences

MDPI

Article

Pilot Sequence Allocation Schemes in Massive MIMO Systems Using Heuristic Approaches

Everton Alex Matos [1], Robson Parmezan Bonidia [2], Danilo Sipoli Sanches [1], Rogério Santos Pozza [1] and Lucas Dias Hiera Sampaio [1,*]

[1] Computer Science Department, Universidade Tecnológica Federal do Paraná, Cornélio Procópio 86300-000, Paraná, Brazil; evertonalexweb@gmail.com (E.A.M.); danilosanches@utfpr.edu.br (D.S.S.); pozza@utfpr.edu.br (R.S.P.)

[2] Institute of Mathematics and Computer Science, Universidade de São Paulo, São Carlos 13566-590, São Paulo, Brazil; bonidia@usp.br

* Correspondence: ldsampaio@utfpr.edu.br

Abstract: This paper presents a comparison of different metaheuristic approaches applied to the pilot sequence allocation problem in Massive Multiple-Input Multiple-Output (MIMO) systems. A modified version of the Genetic Algorithm (GA) as well as different versions of the Particle Swarm Optimization (PSO) Algorithm are used to maximize the system spectral efficiency under an inter-cell interference regime. The metaheuristic parameters were optimized and computational simulations under different scenarios parameters were conducted to verify the system performance impact in terms of system spectral efficiency, minimum and maximum spectral efficiency per user and the cumulative distribution function (CDF) of the users spectral efficiencies. The main contributions of this work are: the creation of a public available dataset; heuristic parameters tuning; findings related to the impact of sub-optimal pilot sequence allocation to the users in terms of maximal and minimal achievable user spectral efficiency and the robustness of some algorithms in scenarios with different system loadings.

Keywords: pilot sequences; resource allocation; Massive MIMO; heuristics

Citation: Matos, E.A.; Parmezan Bonidia, R.; Sipoli Sanches, D.; Santos Pozza, R.; Dias Hiera Sampaio, L. Pilot Sequence Allocation Schemes in Massive MIMO Systems Using Heuristic Approaches. *Appl. Sci.* **2022**, *12*, 5117. https://doi.org/10.3390/app12105117

Academic Editors: Peng-Yeng Yin, Jen-Chun Lee, Hua-Yi Lin, Ming-Chin Chuang, Youcef Gheraibia and Ray-I Chang

Received: 30 March 2022
Accepted: 13 May 2022
Published: 19 May 2022

1. Introduction

The increasing number of multimedia services and online platforms are reflected as a growth in the demand for connectivity and throughput worldwide. According to Cisco [1], in 2018 there were 3.9 billion Internet users globally, and it estimates 5.3 billion in 2023, representing 66% of the global population by that year according to [2]. Part of the global data traffic, 54%, will originate from mobile devices which will be responsible for 160 Exabytes (EB) per month of data traffic in 2025, a 321% growth when compared to the 38 EB per month in 2019 [3].

To support this demand, the wireless communications standard must evolve and increase the total available throughput and the supported number of connected devices, while keeping the average network latency to a minimum. These characteristics will even make new services possible, such as autonomous cars [4].

Massive MIMO is a technology incorporated in 5G networks which corresponds to the transmission of data through several antennas. Thomas L. Marzetta describes the use of this technology for any multi-user MIMO system with more than 16 antennas [5]. The proposed communications scheme relied on low complexity transceivers in the mobile devices and time division duplexing (TDD) to provide a reliable communication channel.

Within a Massive MIMO system, an arbitrary array of bits, known as a pilot sequence, is designated to each user and used in the uplink training process to simplify the signal detection routines in mobile terminals. However, the use of non-orthogonal sequences causes interference among users, which is known as the pilot contamination problem.

To guarantee orthogonality, the size of the pilot sequence must increase with respect to the number of users in the system. Hence, it is easy to verify that pilot sequences must be reused in multi-cell environments and high-density urban areas to avoid long periods of uplink training since as the number of orthogonal sequences increases the size of these sequences also increases.

This work proposes to solve the pilot sequence allocation problem in multi-cell scenarios in a centralized fashion with the maximization of the network spectral efficiency as a goal. The optimization problem is modeled into two different, but equivalent, ways: a binary optimization problem and an integer optimization problem. Meanwhile, to solve these optimization problems, we used different versions of the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) metaheuristics.

1.1. Related Works

Among different studies whose objectives are to minimize the pilot sequence contamination in TDD Massive MIMO systems, there is a considerable selection of methods, tools and techniques already tested, e.g., greedy and tabu search, smart pilot assignment, genetic algorithm (GA), deep learning, hybrid solutions and multi-objective approaches. To situate the contributions of this work, we present a chronological list of related work:

- In [6], the authors approach the pilot contamination on Massive MIMO systems with low complexity evolutionary algorithms comparing the performance in macro-cell scenarios: greedy search, tabu search and a hybrid solution.
- In [7], the authors proposed a new algorithm named Smart Pilot Assignment. The algorithm goal is to maximize all user signals to interference plus noise ratio (SINR) within a cell. The base station calculates the cell observed interference for each of the pilot sequences which users from neighbors' cells cause. The algorithm then assigns the pilot sequences following the rule: the user with the worst channel state receives the pilot sequence with the smallest observed interference. This process is repeated until each user has been assigned a pilot sequence, or the system has no pilot sequences to assign.
- In [8], the authors proposed a greedy search. At each iteration, a number of users are randomly selected from each cell to be assigned the same pilot sequence. The selection is made such that the group chosen maximizes their transmission rate. As the iterations grow, the number of selected users also grows. Likewise, the number of possible selected users at each iteration shortens due to the already accomplished pilot sequence assignments of previous iterations.
- In a different approach [9], the authors proposed the mitigation to pilot sequence contamination modeling the problem as an Epsilon-restricted optimization problem and then solving an eigenvalue decomposition problem through linear complexity.
- In [10], the authors proposed a simple pilot assignment algorithm based on the water-filling algorithm. The users with the best channels received the pilot sequences under the lowest interference effect. To achieve such, at each base station, both channel state information (CSI) array, which contains each user CSI, as well as the inter-cell interference observed by each pilot sequence, are sorted such that one is in increasing order and the other in decreasing order. The pilot assignment is the combination of indexes of these two arrays.
- In [11], the author used different algorithms to solve the pilot sequence assignment problem. The main objective is to maximize the system throughput where the GA was capable of obtaining the average maximum rate user with a smaller complexity. Furthermore, the GA solution as well as random assignment and exhaustive search are compared with the proposed schemes of [6].
- In [12], the authors proposed the assignment of orthogonal pilot sequences along with sectorization. The pilot sequences are re-utilized in the same cell, reducing sizes of the pilot sequence and optimizing spectral efficiency (SE), and Bayesian estimation is used to eliminate pilot contamination.

- In [13], the authors present an adaptive pilot sequence allocation algorithm which separate the users in a cell into two groups: one for users who suffer high interference from other cells and one for users under a low interference regime. The algorithm then assigns mutually orthogonal pilot sequences for all the users under the interference regime, while the other group of users shares the same set of pilot sequences.
- In [14], the pilot sequence allocation problem is solved using deep learning in the form a 3-layer perception neural network. The proposed scheme reaches 99.38% of the theoretical upper-bound performance and takes only 0.92 milliseconds to compute.
- In [15], an algorithm of clustering divides users into two groups, low and high interference. In the group with low interference, the pilots are re-utilized randomly, while high interference groups are grouped by propagation affinity.
- In [16], the authors mitigated the pilot sequence allocation problem through user categorization in high and low interference groups based on large-scale fading, where the high interfering users receive orthogonal pilot sequences, and non-orthogonal pilot sequences are allocated to users under a low interference regime. The authors propose the use of an edge-weighted interference graph to maximize the performance of users in the low interference bracket.
- In [17], the authors proposed the pilot sequence allocation allied to power allocation based on a Monte Carlo Tree Search Method (MCTS) which mitigates the pilot contamination. Moreover, the AlphaGo algorithm is used to play the proposed pilot allocation game, while the Markov Decision Problem (MDP) solves the power allocation problem.
- In [18], based on works [19,20] an adaptation of the particle swarm optimization algorithm to solve the joint pilot sequence allocation and power control problem in Massive MIMO systems was proposed. The authors aim to maximize the spectral efficiency with a limited number of pilot sequences based on coherence interval, while also taking power constraints into account.
- In [21], the authors proposed a joint pilot sequence allocation and antenna scheduling scheme to curb the effects of pilot sequence contamination in Massive MIMO systems where there are a limited number of antennas. To allocate the sequences to multiple users, they proposed rules using either geometric or arithmetic progression in the number of users using the same sequence. Furthermore, they compare their solution with a Greedy pilot sequence allocation scheme and the Smart Pilot Assignment algorithms.
- The work in [22] presents different pilot allocation solutions for cell-free scenarios. The first algorithm is based on the concept of random sequential adsorption using statistical physics, while the second one is an analytical approach of the first one. The authors also describe two centralized algorithms based on clustering principles to benchmark the proposed solutions. The results show that the distributed solutions have a competitive performance compared to the centralized ones, especially when the user density is high.

1.2. Contributions

Our study presented four new main contributions on the subject of Massive MIMO pilot sequence allocation problem:

- The dataset as well as the scripts used to achieve the results are public (Available at github.com/evertonalex/utfpr-ppgi-pilotcontamination, accessed on 16 May 2022);
- The heuristic parameters to the problem assessed were optimized;
- Six different heuristic approaches to the problem comparing it with the simplest solution were evaluated;
- Real scenarios under different parameters to establish the real impact of a pilot allocation scheme were simulated.

Furthermore, we present the differences between our work and those presented in the previous section:

- Two different mathematical models of the same practical problem were addressed;

- Multiple versions and modifications in the PSO algorithm were tested to verify their performance;
- Using The minimum spectral efficiency per user as a performance parameter was used;
- Different scenario parameters were tested to verify the impacts of the pilot sequence schemes in macro, micro and femto-cells.

1.3. Text Organization

This paper is organized as follows. In Section 2, the system model is described with the mathematical problem. In Section 3, the solution with optimization is applied with the evolutionary algorithms. In Section 4, the best parameters are presented and the results of simulations are shown. Finally, in Section 5, we offer the conclusion and final considerations.

2. System Model

We consider a Massive MIMO system with $L > 1$ cells using the same spectrum band and K_ℓ mobile terminals connected with its single base station which has $M >> K_\ell$ antennas. To acquire channel state information (CSI) at each coherence interval, each user in the system must send a pilot sequence through the uplink channel. The pilot sequence uplink signal received by the base station ℓ is a $M \times T_p$ matrix where T_p is the pilot sequence size, and it may be described as:

$$\mathbf{R}_\ell^u = \overbrace{\sum_{k=1}^{K_\ell} \sqrt{p_{k,\ell}}\,\mathbf{g}_{k,\ell,\ell}\,\mathbf{s}_{k,\ell}^H +}^{\text{Desired signal from cell users}} \\ \underbrace{+\sum_{\substack{j=1\\ j\neq\ell}}^{L}\sum_{k'=1}^{K_j}\sqrt{p_{k',j}}\mathbf{g}_{k',j,\ell}\,\mathbf{s}_{k',j}^H}_{\text{Pilot Contamination / Interference}} + \boldsymbol{\eta}_\ell \tag{1}$$

where k, ℓ, k' and j are the user, cell of interest, interfering users and adjacent cell indexers, respectively. Moreover, $\mathbf{R}_\ell^u \in \mathbb{C}^{M\times T_p}$, $p_{k,\ell}$ is the transmission power of each user, $\mathbf{s}_{k,\ell} \in \mathbb{C}^{T_p\times 1}$ is the pilot sequence, whereas $(\cdot)^H$ is the Hermitian operator and is equivalent to the transposed complex conjugate, $\boldsymbol{\eta}_\ell \in \mathbb{C}^{M\times T_p}$ is the noise matrix whose elements are complex Gaussian random variables with zero mean and variance equal to $N_0\,B$ where N_0 is the noise power spectral density (The noise PSD (N_0) is equal to the Boltzmann Constant times the temperature, i.e., it is equivalent to approximately 4.11×10^{-21} watts per hertz at 25 degrees Celsius) and B the system bandwidth. Finally, $\mathbf{g}_{k,j,\ell} \in \mathbb{C}^{M\times 1}$ is the channel gain between the k-th user from cell j and the base station of cell ℓ, which represents the large-scale fading ($\beta_{k,j,\ell}$) and the small-scale fading ($\mathbf{h}_{k,j,\ell}$), i.e.,:

$$\mathbf{g}_{k,j,\ell} = \sqrt{\beta_{k,j,\ell}}\,\mathbf{h}_{k,j,\ell} \tag{2}$$

where $\mathbf{h}_{k,j,\ell} \in \mathbb{C}^{M\times 1}$ are independent and identically distributed complex Gaussian random variables with zero mean and unit variance. Meanwhile, $\beta_{k,j,\ell}$ is the path loss and shadowing effects. We assume a simplified path loss model, i.e.,:

$$\beta_{k,j,\ell} = \left(\frac{\lambda}{4\,\pi\,d_0}\right)^2 \left(\frac{d_0}{d_{k,j,\ell}}\right)^\gamma \mathcal{S}_{k,j,\ell} \tag{3}$$

where λ is the wavelength, d_0 is the reference distance (Typically, $d_0 \in [1, 10]$ meters for indoor environments and $d_0 \in (10, 100)$ meters for outdoors), $d_{k,j,\ell}$ is the distance (in meters) from the k-th user of cell j to the base station in cell ℓ. Finally, $\gamma \in [2, 8]$ is the path loss exponent which is directly related to the scenario where the wireless communications take place, and $\mathcal{S}_{k,j,\ell}$ is the shadowing log-normal distributed random variable with zero mean and variance $10^{\frac{\sigma_s^2}{10}}$ where $\sigma_s^2 \in [4, 13]$ for outdoor channels.

We suppose the use of an orthogonal variable spreading factor (OVSF) code to generate the pilot sequences which are designated to each user through the pilot sequence allocation hipermatrix $\mathbf{\Phi} \in \{0,1\}^{K_\ell \times T_p \times L}$ whose elements are defined as:

$$\phi_{k,q,\ell} = \left\{ \begin{array}{ll} 0 & \text{pilot sequence is not allocated} \\ 1 & \text{pilot sequence is allocated} \end{array} \right\} \tag{4}$$

Moreover, according to [5], the uplink signal to interference plus noise ratio (SINR) of user k from cell ℓ observed by its base station when the number of antennas grows indefinitely is:

$$\delta^{\mathbb{B}}_{k,\ell} = \frac{\beta_{k,\ell,\ell}}{N_0\, B + \sum\limits_{\substack{j \neq \ell \\ j=1}}^{K_j} \sum\limits_{k'=1}^{K_j} \boldsymbol{\phi}_{k,\ell} \boldsymbol{\phi}^T_{k',j}\, \beta_{k',j,\ell}} \tag{5}$$

where $\boldsymbol{\phi}_{k,\ell}$ is a row from the hipermatrix $\mathbf{\Phi}$, i.e., it is a $\{0,1\}^{1 \times T_p}$ array that designates which pilot sequence is allocated to user k of cell ℓ. Whenever $\boldsymbol{\phi}_{k,\ell}\boldsymbol{\phi}^T_{k',j} = 1$, users k and k' from cell ℓ and j, respectively, use the same pilot sequence and, therefore, interfere in each others signal.

Alternative Representation

An alternative to the allocation variable binary representation is to use an integer representation, i.e., instead of using a binary pilot sequence allocation hipermatrix for the set of cells, one may use a single integer pilot sequence allocation $\mathbf{\Theta} \in \{1, \ldots, T_p\}^{L \times K}$ such that:

$$\theta_{\ell,k} = x \tag{6}$$

if the pilot sequence number $x \in \{0, \ldots, T_p\}$ is allocated to user k of cell ℓ.

In this representation, we introduce a new function which is responsible to verify if user k' from cell j is using the same pilot sequence as user k from cell ℓ. Therefore, consider $F(k, \ell, k', j) : \mathbb{Z}^{K \times L \times K \times L} \to \{0,1\}$ such that:

$$F(k, \ell, k', j) = \left\{ \begin{array}{ll} 0, & \text{if } \theta_{\ell,k} \neq \theta_{j,k'} \\ 1, & \text{otherwise} \end{array} \right. \tag{7}$$

Furthermore, we may rewrite Equation (5) using (7):

$$\delta^{\mathbb{Z}}_{k,\ell} = \frac{\beta_{k,\ell,\ell}}{N_0\, B + \sum\limits_{\substack{j \neq \ell \\ j=1}}^{K_j} \sum\limits_{k'=1}^{K_j} F(k, \ell, k', j)\, \beta_{k',j,\ell}} \tag{8}$$

It is clear from Equations (5) and (8) that we may derive two equivalent optimization problems which are described in the next section.

3. Optimization Problems

The pilot sequence allocation optimization problem discussed here aims to maximize the total system spectral efficiency, simultaneously satisfying the constraints of pilot sequence orthogonality within the same cell as well as the fact that no users should be assigned different pilot sequences at the same time.

Moreover, it is important to differentiate the two mathematical models of allocation into two optimization problems: one optimization problem (OP1), as in Equation (9), uses the binary allocation matrix representation found in Equation (5) and is mathematically defined as:

$$\begin{aligned}
\underset{\mathbf{\Phi}}{\text{maximize}} \quad & \mathcal{J}_1(\mathbf{\Phi}) \;=\; \sum_{\ell=1}^{L}\sum_{k=1}^{K_\ell}\log_2\left(1+\delta_{k,\ell}^{\mathbb{B}}\right) && (9)\\
\text{subject to} \quad & \sum_{q=1}^{T_p}\phi_{k,q,\ell} \leq 1,\ \forall\, k \text{ and } \ \ell && (10)\\
& \sum_{k=1}^{K_\ell}\phi_{k,q,\ell}\!= 1,\ \forall\, q \text{ and } \ \ell && (11)\\
& \phi_{k,q,\ell} \in \{0,1\},\ \forall\, k,\ q,\ \ell && (12)
\end{aligned}$$

where Equation (10) assures that no more than one pilot sequence is assigned to each user, while Equation (10) guarantees that each pilot sequence is used by only one user in each cell, and Equation (12) imposes that the decision variable is binary. Meanwhile, optimization problem 2 (OP2), as in Equation (13), uses the integer index representation introduced in (8) and may be described as:

$$\begin{aligned}
\underset{\mathbf{\Theta}}{\text{maximize}} \quad & \mathcal{J}_2(\mathbf{\Theta}) \;=\; \sum_{\ell=1}^{L}\sum_{k=1}^{K_\ell}\log_2\left(\delta_{k,\ell}^{\mathbb{Z}}\right) && (13)\\
\text{subject to} \quad & \theta_{\ell,k} \neq \theta_{\ell,k'},\ \forall\, \ell,\, k \text{ and } k' && (14)\\
& \theta_{\ell,k} \in \{1,\ldots,T_p\},\ \forall\, \ell \text{ and } k && (15)
\end{aligned}$$

where Equation (10) assures that the same pilot sequence is used more than one time within a cell, avoiding intra-cell interference, while Equation (11) assures that only the available pilot sequences are allocated to each user in the cell. Note that the constraints in OP2 are an integer adaptation of the same constraints in OP1.

To solve the optimization problem in (9), we use two different metaheuristic approaches. First, the binary version of the Particle Swarm Optimization (PSO) algorithm is considered. Later, a binary version of the Genetic Algorithm (GA) is also used to achieve a centralized pilot sequence allocation. Furthermore, we use the Smallest Position Value (SPV) technique along the PSO algorithm and Variable Neighbourhood Search (VNS) to solve the second optimization problem in (13). It is worth noting that although (9) and (13) have different domains, they represent the same physical problem and the conversion from one domain to another follows the rule presented by Equation (6).

4. Proposed Solutions

The optimization problems are solved using different methods: binary PSO (BPSO) and GA to solve (9), while PSO, SPV and VNS are used to solve (13). In this section we provide a detailed explanation of how these techniques are applied in this context.

4.1. Binary Particle Swarm Optimization

The BPSO was first described by Kennedy and Eberhart [23] and was an adaptation of the original (continuous) PSO. This algorithm works based on the behaviour of birds in search of foods, and the each candidate solution is presented as a bird in a flock flying through the search space (optimization problem domain), and the found represents the optimal solution. At each algorithm iteration, the velocity of each particle is updated through the equation:

$$\mathbf{v}_i[t+1] = \omega\,\mathbf{v}_i[t] + c_1\,r_1\,(\mathbf{x}_i[t]-\mathbf{p}_i) + c_2\,r_2\,(\mathbf{x}_i[t]-\mathbf{p}_g) \tag{16}$$

where ω is the inertia weight, $\mathbf{v}_i$ is the candidate i velocity, $\mathbf{x}_i$ is the particle position in the search space, $\mathbf{p}_i$ is the best position candidate i has ever been to in terms of fitness function, and $\mathbf{p}_g$ is the best position overall the whole population, i.e., the first keeps an individual best position record and the latter, a global best position record. The coefficients c_1 and c_2 are the cognitive and social acceleration constants, respectively, whilst r_1 and r_2 are uniformly distributed random numbers in the interval $[0, 1]$.

In a classical PSO algorithm, the next step would be updating each candidate position using its velocity. However, to implement a binary version, the change in each candidate dimensions must also be either zero or one. Therefore, a sigmoid function is applied to each velocity component, and it is equivalent to the probability of changing each bit. In this paper, we use the following sigmoid function in each dimension $\mathbf{v}_i$:

$$\mathcal{S}(v_i[t]) = \frac{1}{1 + e^{-v_i[t]}} \tag{17}$$

The sigmoid function value is then compared to a random value generated to each dimension of the candidate $\mathbf{x}_i$ such that:

$$x_i[t+1] = \begin{cases} 1 & \text{if } r < \mathcal{S}(v_i[t]) \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where $r \sim \mathcal{U}(0, 1)$, i.e., r is a random variable with uniform distribution over the interval $(0, 1)$.

Although the position possibilities are constrained to $\{0, 1\}$, particle velocity may grow indefinitely. Hence, a maximum velocity constraint may be applied to each dimension, i.e.,:

$$v_i[t+1] = \begin{cases} v_{\max} & \text{if } v_i[t+1] > v_{\max} \\ -v_{\max} & \text{if } v_i[t+1] < -v_{\max} \end{cases} \tag{19}$$

Furthermore, since OP1 in (9) is a constrained optimization problem, we present two alternatives to treat the unfeasible solution candidates. The first is simply discarding the unfeasible candidates and replacing them with a random feasible one; the second is using a new fitness function defined as:

$$\tilde{\mathcal{J}}_1(\mathbf{\Phi}) = \begin{cases} \mathcal{J}_1(\mathbf{\Phi}) & \text{if feasible} \\ \mathcal{J}_{\min} - \mathcal{P} & \text{if unfeasible} \end{cases} \tag{20}$$

where $\mathcal{J}_{\min}$ is the worst feasible solution in the PSO population, and $\mathcal{P}$ is the sum of the constraints in Equations (10) and (11), i.e.,:

$$\begin{aligned} \mathcal{P} \quad = \quad & \sum_{\ell=1}^{L} \sum_{k=1}^{K_\ell} \left(\sum_{q=1}^{T_p} \phi_{k,q,\ell} - 1 \right) + \\ & \sum_{\ell=1}^{L} \sum_{q=1}^{T_p} \left(\sum_{k=1}^{K_\ell} \phi_{k,q,\ell} - 1 \right) \end{aligned} \tag{21}$$

Finally, a pseudocode of the BPSO algorithm is presented in Algorithm 1.

Algorithm 1: BPSO

```
input  : Channel gain Hipermatrix—$\boldsymbol{\beta}$
         Population Size—$N$
         Max. Number of Iterations—$I$
output : Pilot Sequence Allocation Hipermatrix $\boldsymbol{\Phi}$
Create $N$ random candidate solutions $\mathbf{x}_1[1], \ldots, \mathbf{x}_N[1]$;
Evaluate $\mathcal{J}(\mathbf{x}_1), \ldots, \mathcal{J}(\mathbf{x}_N)$ acoording to (9);
Start $\mathbf{p}_i$ for each $i = 1, \ldots, N$ and $\mathbf{p}_g = \arg\max_{\mathbf{p}_i} \mathcal{J}(\mathbf{p}_i)$;
for $i = 1$ until $I$ do
    for $j = 1$ until $N$ do
        Update $\mathbf{x}_j[i+1]$ using Equation (16) up to Equation (19);
        Evaluate the new solutions using Equations (20) and (21);
        if $\mathcal{J}(\mathbf{x}_j[i+1]) > \mathcal{J}(\mathbf{p}_j)$ then
            $\mathbf{p}_j = \mathbf{x}_j[i+1]$;
        end
        if $\mathcal{J}(\mathbf{x}_j[i+1]) > \mathcal{J}(\mathbf{p}_g)$ then
            $\mathbf{p}_g = \mathbf{x}_j[i+1]$;
        end
    end
end
```

4.2. PSO-Smallest Position Value

To solve the optimization problem (13), we used the Smallest Position Value method along with the PSO algorithm (PSO–SPV). In the SPV, the value of each problem dimension is exchanged by the index of the sorted values. In this specific problem, it is equivalent to the SPV value designating the pilot sequence allocated to each user. The sorting process which is responsible for swapping the real values for their ordinal ones is performed separately for each cell in the system. To illustrate the application of the SPV method, we present Table 1 which relates the value of each PSO particle (Real Value) to the pilot sequence assignment when four dimensions (users) are accounted for in a single cell.

Table 1. Example of operation of SPV in a single cell, $K = 4$ users scenario.

Dimension (User)	1	2	3	4
Real value	4.85	−2.15	145	−1.333
SPV (Pilot Sequence)	3	1	4	2

Using an SPV changes the problem domain from a binary scenario to integers values such that $\theta_{k,\ell} = x$ where k is the dimension of the SPV scheme and x is the SPV value, while ℓ designates the cell of interest.

The PSO is used along with the SPV through the application of Equation (16) and the update position equation:

$$\mathbf{x}_i[t+1] = \mathbf{x}_i[t] + \mathbf{v}_i[t+1] \tag{22}$$

It is important to point out that the velocity limits defined by Equation (19) are also applied to the PSO–SPV solution. Furthermore, from the nature of this approach, it is unlikely that two users will have the same pilot sequence assigned, or the other way around, since the sorting process would only place two users in the same position if their real values are exactly the same. To avoid this situation, in case two dimensions of an individual present the exact same real value, the tie break is the user index: the smallest user indexer is sorted as the smallest value. Due to these two facts, the PSO–SPV fitness function is the objective function presented in (9), which means that the BPSO and PSO–SPV use different objective functions although the algorithms objective is the same: maximizing the system spectral efficiency.

4.3. Variable Neighbourhood Search

Another technique we also used to make the PSO–SPV approach even more robust is the Variable Neighbourhood Search metaheuristic which is based on the systemic change of neighborhoods of each possible solution in an attempt to find better solutions near the PSO candidate solutions.

VNS is applied to the PSO–SPV through an exchange of sequences allocated to user pairs on the same cell, i.e., users are randomly paired, and their pilot sequences are swapped whenever VNS is applied at each iteration of the PSO–SPV algorithm. Mathematically for each pair k' e k'' of users:

$$\begin{aligned} \theta_{k',\ell} = q' \quad &\rightarrow \quad \theta_{k',\ell} = q'' \\ \theta_{k'',\ell} = q'' \quad &\rightarrow \quad \theta_{k'',\ell} = q' \end{aligned} \tag{23}$$

where q' and q'' are the assigned pilot sequences. It is worth noting that when the number of the users is even, one of the users is left out of the VNS subroutine, and its pilot sequence remains unchanged.

To avoid a large increase in complexity, this mechanism is not performed for all individuals of the PSO–SPV population. In fact, we define r_{vns} as the probability of running VNS for each candidate solution in each one of the PSO–SPV iterations. The best balance between complexity and solution quality for the r_{vns} parameter is discussed in the results section.

Finally, the PSO–SPV and PSO–SPV–VNS algorithms are presented in the Algorithm 2.

Algorithm 2: PSO–SPV–VNS

```
input  : Channel gain Hipermatrix—$\boldsymbol{\beta}$
         Population Size—$N$
         Max. Number of Iterations—$I$
output : Pilot Sequence Allocation Hipermatrix $\boldsymbol{\Theta}$
Create $N$ random candidate solutions $\mathbf{x}_1[1], \ldots, \mathbf{x}_N[1]$ in the $\mathbb{R}^{K \times L}$ space;
Apply SPV to each $\mathbf{x}_i[1]$ generating the pilot sequence allocation array $\boldsymbol{\theta}_i[1]$;
Evaluate $\mathcal{J}(\boldsymbol{\theta}_i[1]), \ldots, \mathcal{J}(\boldsymbol{\theta}_N[1])$ according to (13);
Start $\mathbf{p}_i$ for each $i = 1, \ldots, N$ and $\mathbf{p}_g = \arg\max_{\mathbf{p}_i} \mathcal{J}(\mathbf{p}_i)$;
for $i = 1$ until $I$ do
    for $j = 1$ until $N$ do
        Update $\mathbf{x}_j[i+1]$ using Equations (16), (19) and (22);
        Apply SPV to each $\mathbf{x}_i[t+1]$ generating the pilot sequence allocation array
          $\boldsymbol{\theta}_i[i+1]$;
        if rand() $< r_{\text{vns}}$ then
            Randomly pair users and performe VNS through (23);
        end
        if $\mathcal{J}(\boldsymbol{\theta}_j[i+1]) > \mathcal{J}(\mathbf{p}_j)$ then
            $\mathbf{p}_j = \boldsymbol{\theta}_j[i+1]$;
        end
        if $\mathcal{J}(\boldsymbol{\theta}_j[i+1]) > \mathcal{J}(\mathbf{p}_g)$ then
            $\mathbf{p}_g = \boldsymbol{\theta}_j[i+1]$;
        end
    end
end
```

4.4. Genetic Algorithm

GA is an evolutionary computation method created by John H. Holland [24], wherein the algorithm simulates the theory of evolution by Charles Darwin.

The process consists of taking a population of individuals which are composed of chromosomes and applying different genetic operations to the individuals chromosomes, combining them while aiming to improve some fitness function. At each iteration of the algorithm, called a generation, individuals pair up to create descendants creating a new generation. During this process, the chromosomes, which can represent a position in the problem domain, may suffer mutation, while the individuals derived from their parents are created from a crossover operation. At each generation, only the best suitable individuals survive and transmit their genes to the next generation.

To further specify how the GA is applied to the optimization problem, we define the fitness function as the function $\mathcal{J}$ in (9). In addition, the GA will have three operators: a proportional fitness selection to define the parents for each new individual, the crossover operator and the the mutation operator.

The first step in each iteration is to select the parents for the next generation. Here we apply a fitness proportional selection, i.e., the probability of each individual i at some generation t to be selected as a parent is proportional to $\mathcal{J}(\mathbf{x}_i[t])$. Mathematically, the probability of i being selected as a parent is defined as:

$$\Pr(\boldsymbol{\Phi}_i) = \frac{\mathcal{J}(\boldsymbol{\Phi}_i)}{\sum\limits_{j=1}^{N} \mathcal{J}(\boldsymbol{\Phi}_j)} \tag{24}$$

The crossover operator is responsible for combining the parent's chromosomes and generating the children's ones. To do this and simultaneously satisfy the constraints in Equations (10)–(12), we propose the use of a one-point crossover. To understand how the crossover operation is carried out, we first revisit the pilot sequence allocation hypermatrix which is a $K \times T_p \times L$ hypermatrix. Moreover, the crossover point is set at one of the cells and is defined as ℓ'. Hence, the first child is made of the individual allocation matrices from both parents: from cell 1 to ℓ', the matrices are from the first parent and from cell $\ell' + 1$ to L from the second parent. Its sibling is the mirror of that, i.e., the first parent passes the matrices from $\ell' + 1$ to L, while the second parent the matrices from 1 to ℓ. We define $\ell' = \lfloor L/2 \rfloor$.

The choice of this technique among the many others in the literature is involved in creating children to AG where they did not respect all the constraints of the problem domain. For this reason, the one-point crossover utilized granted that all children generated of crossing fathers are on the feasible solution for the problem.

To preserve the best solution throughout the generations if all the descendants at a given generation are worse than the best individual so far, we replace the worst individual of that generation with the best solution from the previous generation.

Finally, the mutation of an individual is carried out in a way that also satisfies the constraints in Equations (10)–(12). Without the loss of generality, let $K = T_p$ be even. The first step of the mutation operator is to generate a random permutation of the numbers from one to K for each cell ℓ. Let $\mathbf{p}_\ell = [p_1, \ldots, p_K]$ be that permutation, the first half of the permutation $\mathbf{p}_\ell$, i.e., from p_1 to $p_{\frac{K}{2}}$, is paired up element-wise with the second half, that is $p_{\frac{K}{2}+1}$ to p_K. Finally, we perform operations in the child hypermatrix such that for each cell's permutation the users defined by the number in each half permutation swap their pilot sequence. It is worth noting that mutation does not happen for every individual since there is a mutation rate $T_m \in (0, 1)$ which defines the probability of a mutation occurring in each member of a population.

These three operations happen at each iteration of the GA and are repeated until the maximum number of generations is achieved. The pseudocode for the adapted binary GA is presented in Algorithm 3.

Algorithm 3: Binary Genetic Algorithm

```
input  : Max number of generations—$G_{\max}$, Mutation ratio—$T_m$
         Channel gain hypermatrix—$\boldsymbol{\beta}$, Population size—$N$
output :
         Hypermatrix of allocated pilot sequences—$\boldsymbol{\Phi}$
Create $N$ candidate solutions $\mathbf{x}_1[1], \dots, \mathbf{x}_N[1]$;
for $t = 1$ until $G_{\max}$ do
    for $n = 1$ until $N/2$ do
        Select two parents $\mathbf{x}_i[t]$ and $\mathbf{x}_j[t]$ using proportional selection;
        Perform the crossover operation generating $x_i[t+1]$ and $x_j[t+1]$;
        if $rand(0,1) > T_m$ then
            Perform the mutation operation;
        end
    end
    Evaluate $\mathcal{J}(\mathbf{x}_n[t+1])$ for all $x$;
    if $\nexists\, n$ such that $\mathcal{J}(\mathbf{x}_n[t+1]) > \max(\mathcal{J}(\mathbf{x}[t]))$ then
        Preserve the best solution from the generation $t$ replacing the worst
          solution in $t+1$:;
        $\min_n(\mathcal{J}(\mathbf{x}_n[t+1])) = \max_n(\mathcal{J}(\mathbf{x}_n[t]))$ ;
    end
end
```

The literature has other candidate solutions for use, but this problem as described in Section 2 is a combinatorial optimization problem. In a general way, the methods utilized are mathematically complex and in binary or integer scenarios. Considering all the constraints introduced, the approach in the literature is not easy to adapt.

5. Simulations Results

We verify the applicability of the techniques presented in this manuscript to the pilot sequence assignment problem, as well as the impact of using them in terms of user SINR, system spectral efficiency and minimum and maximum user spectral efficiency. We developed a MATLAB script to create a dataset and made it public at github.com/evertonalex/utfpr-ppgi-pilotcontamination, accessed on 12 May 2022.

We consider a seven hexagonal cell cluster geographically disposed such as presented in Figure 1. Each user in the cell is randomly positioned. However, to establish some physical limits, user positions can only be integers numbers pairs and must be contained within the limits of the hexagon, i.e., let $(x, y) \in \mathbb{Z}^2$ be some user position related to its base station which is fixed at the origin, $x, y \sim \mathcal{U}\{1, R\}$, and the following conditions are also met:

1. $|x| \leq R$
2. $|y| \leq \frac{\sqrt{3}}{2} R$
3. $\frac{\sqrt{3}}{2} |x| + |y| \geq \frac{\sqrt{3}}{2} R$

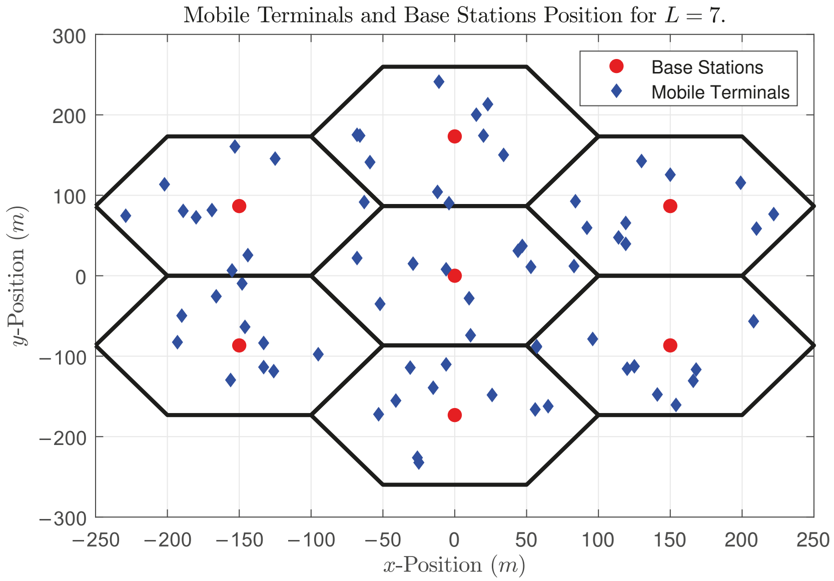


Figure 1. Base stations, users and cells geographic disposition in our simulation scenario. User positions were randomly generated.

After their positions are randomly generated with respect to the origin, each user position is translated to their cell position. Moreover, this process replaces any users that overlap the same position with a new random one. Table 2 shows the parameters used in the aforementioned scenario.

Table 2. System parameters used in the computational simulations.

Parameter	Adopted Value(s)
Number of Cells (L)	7
Number of Users per Cell (K)	$\{20, 40, 60\}$
Cell Radius (R)	$\{100, 250, 500, 1000\}$ (m)
Wavelength (λ)	8.56 (cm)
Reference Distance (d_0)	10 (m)
Path Loss Exponent (γ)	$\{6, 2\}$
Shadowing Variance (σ_s^2)	$\{6, 10\}$ (dB)
Channel Bandwidth (B)	20 (MHz)
Transmission frequency	3.5 GHz
Dataset instances	1000 per set of parameters

5.1. Heuristic Parameters Optimization

At this point, when the performance results are combined with the execution time results, such as observed in Figure 2 and 3, more iterations do not represent a more significant increase in performance. Hence, the best approach is to use as few iterations as possible in each of the algorithms due to the high cost and low increase in performance. All the simulations use the optimal parameters for each heuristic, and the number of iterations is set to 50.

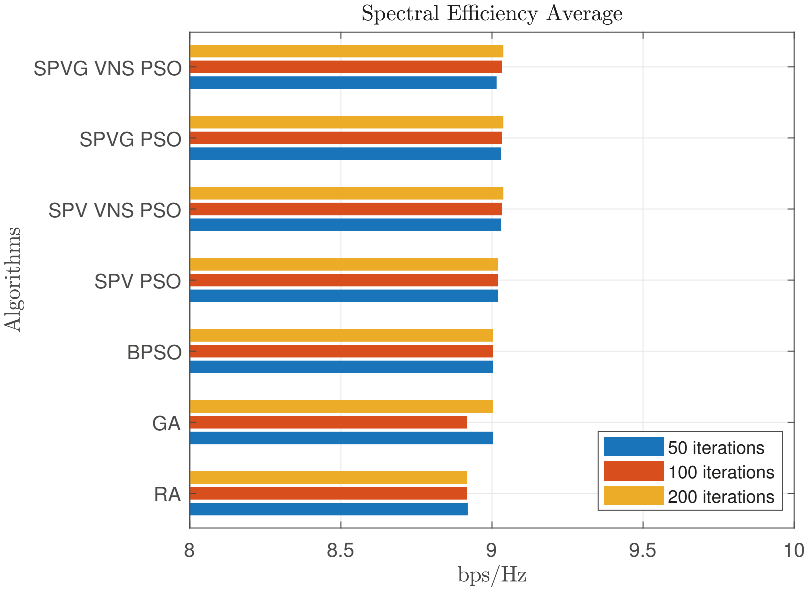


Figure 2. Spectral efficiency average by iterations.

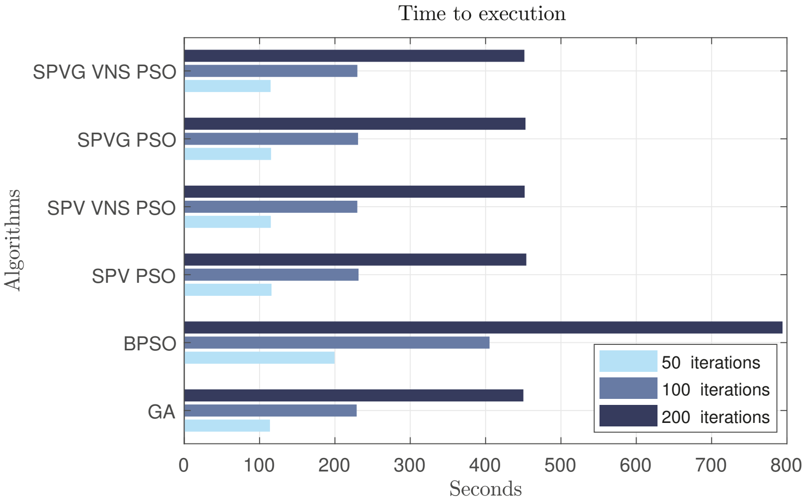


Figure 3. Time to complete the execution of a single instance in the dataset by each algorithm. We used a Google Cloud solution with 32 virtual CPUs, 32GB of RAM, running MATLAB on Ubuntu 20.

Since each heuristic has its own set of parameters which may impact the overall performance, we investigated the algorithms performance in the following scenario: $K = 20$ users and $R = 1000$ m. The findings, for each heuristic, are presented in Table 3. The values which result in the best performance are marked in bold type.

Table 3. Set of tested heuristic parameters. The parameters in bold are the ones which resulted in the best system spectral efficiency.

BPSO Parameters	Tested Value(s)
Inertia (ω)	$\{0.1, 0.25, \mathbf{0.5}\}$
Max. Velocity $(v_{\max})$	$\{1, 2, \mathbf{3}\}$
Coefficient (c_1, c_2)	$\{(\mathbf{1}, \mathbf{1}), (2, 2), (3, 3)\}$
Max. # of Iterations (I)	$\{50, 100, \mathbf{200}\}$
Population Size (N)	$2 \times K$
SPV(G)-PSO Parameters	**Tested Value(s)**
Variable neighborhood search (VNS)	$\{0.15, 0.30, 0.60\}$
Max. # of Iterations (I)	$\{50, 100, \mathbf{200}\}$
Population Size (N)	$2 \times K$
GA Parameters	**Tested Value(s)**
Mutation Rate (T_m)	$\{0.25,\ 0.50,\ \mathbf{0.75}\}$
Max. # of Generations $(G_{\max})$	$\{50, 100, \mathbf{200}\}$
Population Size (N)	$2 \times K$

It is worth noting that although we tested different parameters for each heuristic, the results in terms of system spectral efficiency are where all are within a less than 1% margin from each other. For instance, the increment in the results for GA when $G_{\max}$ goes from 100 to 200 is around 0.0019%. Additionally, the increment from using a 0.5 mutation rate to a 0.75 is around 0.0067%. In both cases, the difference means less than a bit per second per Hertz, which mean they are irrelevant system-wise.

This pattern is also observed in the BPSO, SPV–PSO and SPVG–PSO (with or without VNS) results. The difference between the parameters is within the margin of error for the dataset size used in our work.

Since the difference between the algorithms performance in terms of average SE is marginal, we further analyzed the minimal spectral efficiency case. The argument here is that increasing the minimal SE may result in a reduction in terms of outage probability.

Thus, Table 4 presents the minimal SE for different combinations of maximum number of iterations and for each algorithm. It is worth noting that only the optimal heuristic parameters were used in this simulation. Furthermore, it is clear that the tested heuristics are divided into two groups when we compare their performance: the low performing group includes RA, BPSO and GA while the high performing group is composed of SPV–PSO, SPV–VNS–PSO, SPVG–PSO and SPVG–VNS–PSO.

Table 4. Minimal spectral efficiency (in bps/Hz) using the optimal parameters for each heuristic ($\omega = 0.25$, $v_{\max} = 2$, $VNS = 0.30$, $T_m = 0.50$ and $K = 20$).

Algorithm	Max. # of Iterations (I)		
	50	**100**	**200**
RA	7.56×10^{-5}	7.56×10^{-5}	7.56×10^{-5}
BPSO	7.56×10^{-5}	7.56×10^{-5}	7.56×10^{-5}
GA	7.56×10^{-5}	7.56×10^{-5}	7.56×10^{-5}
SPV PSO	1.4143	1.4329	1.4246
SPV VNS PSO	1.3762	1.3364	1.3239
SPVG PSO	1.3897	1.3794	1.3956
SPVG VNS PSO	1.3191	1.3419	1.2982

Besides the aforementioned fact, one can also imply that the average variation in performance among all algorithms in the high performance bracket when the number of iterations is increased from 50 to 100 is less than 2%. Additionally, even though Table 4 shows that RA, BPSO and GA had the same performance despite the change in the number of iterations, the average difference in performance was in the order of 0.0013% when the number of iterations rise from 50 to 100.

Finally, as it is commonplace, increasing the number of iterations directly impacts the computational time of all the algorithms but RA. Hence, we present Figure 3 which shows the average time needed to compute the solution.

At this point, it is clear when the performance results are combined with the execution time results that the best approach is to use as few iterations as possible in each of the algorithms due to the high cost and low increase in performance. From now on, all the simulations use the optimal parameters for each heuristic, and the number of iterations is set to 50.

As one may observe, each algorithm has a set of parameters that need to be adapted to the discussed problem. In order to find the best parameters, we considered a macro-cell scenario scenario with $K = 20$ users per cell of $R = 1000$ m in radius and a maximum number of iterations $I = 50$ to all heuristics.

For the PSO-based algorithms, we first analyzed the inertia coefficient ω. Figure 4 shows the results which demonstrate that $\omega = 0.25$ is the best configuration for both BPSO and SPVG–PSO, while $\omega = 0.1$ is the best option for SPV–PSO. Even though the performance differences of the tested parameters values are marginal, the best values were used to evaluate other aspects of the proposed solutions.

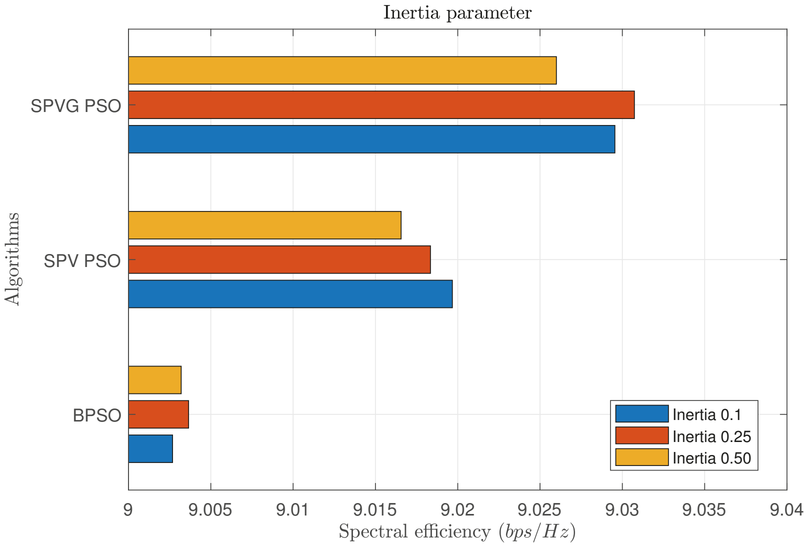


Figure 4. Results of tests to find the better parameter to inertia ($K = 20$, $R = 1000$ m and $v_{\max} = 1$).

The second parameter tested in the PSO-based algorithms was the maximum velocity $(v_{\max})$. Values from one to four were used, and the best results were different for each algorithm: $v_{\max} = 4$ for the BPSO, $v_{\max} = 3$ to SPV–PSO and $v_{\max} = 2$ to SPVG–PSO. The tests results are shown in Figure 5. As it was observed with the inertia coefficient, the increment in system performance were less than 0.1% when compared to the random allocation process.

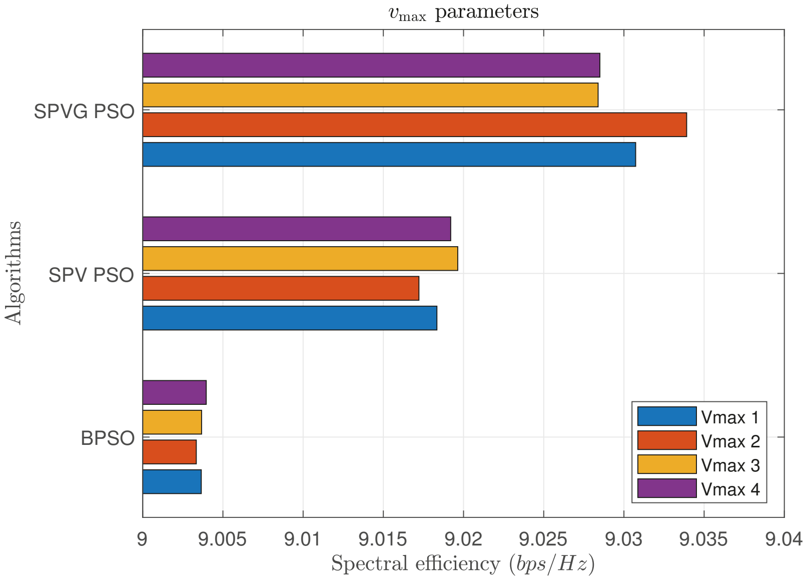


Figure 5. Result of tests to find the better parameter to $v_{\max}$ ($K = 20$, $R = 1000$ m and $(\omega) = 0.25$).

The third and forth parameters tested were the local and global solution weights (c_1 and c_2). These parameters are not found in the SPVGs variants, so they were only tested for BPSO and SPV–PSO.

All the previous analysis considered that $c_1 = c_2 = 1$, so we compared the results with those solutions. Increasing the local solution weight $c_1 = 2$ and maintaining the global weight $c_2 = 1$ downgrades the BPSO performance in $0,001\%$, while increasing the SPV–PSO performance in 0.02%. Further increasing the local solution $c_1 = 5$ and $c_2 = 1$ made the BPSO algorithm's performance around 0.004% higher than the baseline ($c_1 = c_2 = 1$), while an increase of 0.03% was found for the SPV–PSO when compared to the baseline. Finally, changing the parameters to $c_1 = 1$ and $c_2 = 5$ increased the BPSO performance in 0.005% while decreased the SPV–PSO performance in 0.02%

This test showed that c_1 and c_2 values are not significant in terms of performance since their impact are in the one thousandth of a bit/s/Hz mark.

The last PSO-based parameters optimization was the VNS rate. We tested four different configurations, and the results are presented in Figure 6. They demonstrate that a rate of 0.9 is the best configuration for both SPV–PSO–VNS and SPVG–PSO–VNS.

Finally, the GA algorithm parameter was tested. Figure 7 shows the GA's performance for different mutation rates, and the $T_m = 0.9$ was found to be the best parameter value.

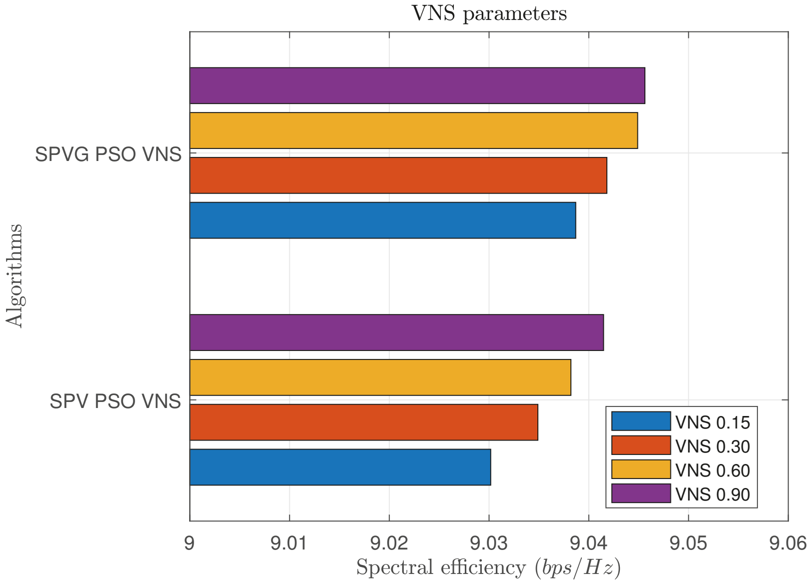


Figure 6. Results of tests to find the better parameters to VNS ($K = 20$, $R = 1000$ m, $(\omega) = 0.25$ and $v_{\max} = 1$).

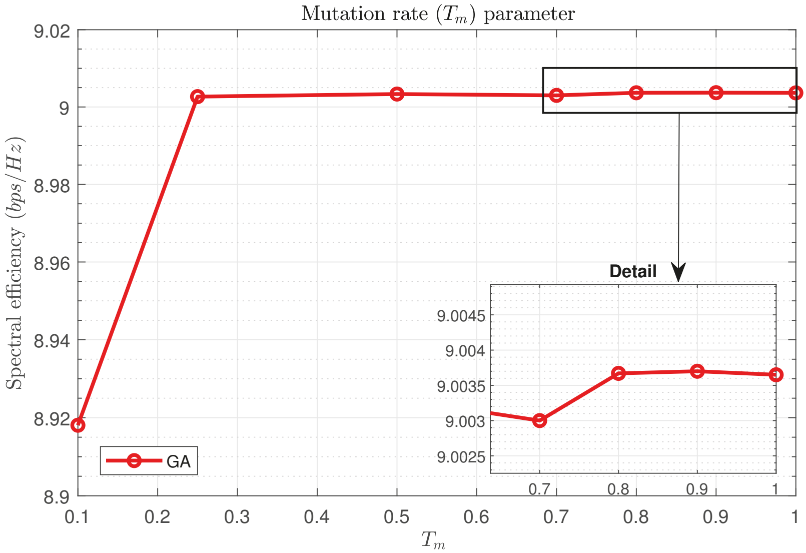


Figure 7. Results of tests to find the better parameters to T_m ($K = 20$, $R = 1000$ m).

5.2. Cell Size Impact on Performance

We conducted an analysis of the impact of cell size on all of the algorithms performance. We set up a scenario with $K = 20$ users per cell and the maximum number of iterations per algorithm to 50. To evaluate performance, Figure 8 shows the cumulative distribution function using our dataset.

As one may see, the impact of cell size on the CDF is really marginal, which may indicate that the system is not interference bounded. This is corroborated with two facts.

No intracell interference exists due to the orthogonality of the pilot sequences within a cell. The path-loss and shadowing effects deteriorate the interfering signals from other cells.

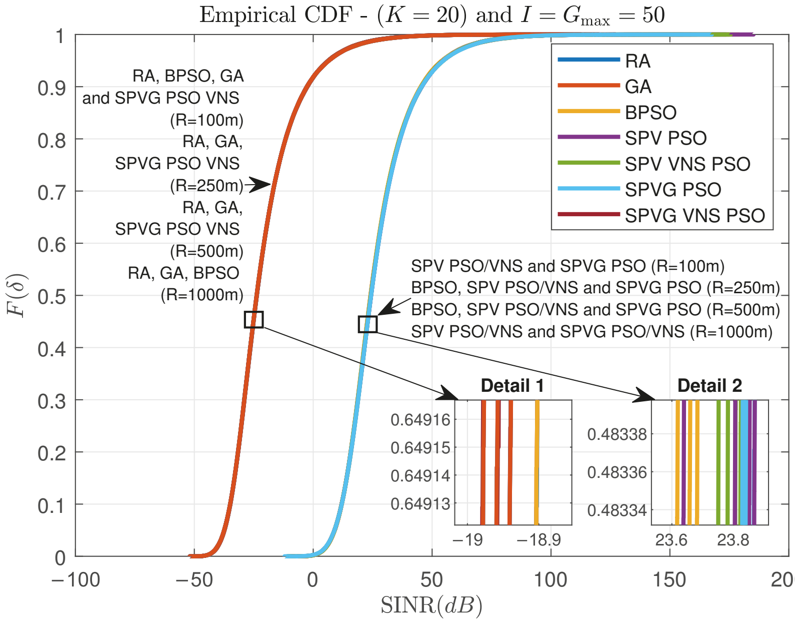


Figure 8. Empirical CDF for different cell sizes with $K = 20$ users, $G_{\max} = 50$ generations and $I = 50$ iterations.

Although the optimization problem aims to maximize the system spectral efficiency, we analysed the minimum spectral efficiency among all users in the system which may be seen as a fairness measure in the system. Table 5 presents the numerical values of the average lowest spectral efficiency in the system considering the 1000 instances in the dataset. The difference between the results for the RA, BPSO and GA algorithms were marginal, so they are listed in the same column.

An important remark to make here is: for a cell with a 100 m radius, the SPV–PSO has an average lowest spectral efficiency 1.7% higher than the RA, BPSO and GA algorithms.

Table 5. Minimum spectral efficiency per user in different cell sizes. The best value for each algorithm is marked in bold type.

Parameters	Minimum Spectral Efficiency per User (in bps/Hz)				
	BPSO/GA	SPV–PSO		SPVG–PSO	
		w/o VNS	with VNS	w/o VNS	with VNS
$K = 20, R = 1$ Km	7.56 $\times 10^{-5}$	1.4143	1.3762	1.3897	1.3191
$K = 20, R = 500$ m	7.85 $\times 10^{-5}$	1.4233	**1.402**	**1.4093**	1.3561
$K = 20, R = 250$ m	7.90 $\times 10^{-5}$	1.4142	1.3858	1.3782	**1.3873**
$K = 20, R = 100$ m	**8.03** $\times 10^{-5}$	**1.4414**	1.3605	1.3986	1.3424

In relation to the average maximum spectral efficiency, the results are presented in Table 6. It is clear that RA, BPSO and GA algorithms have a worse performance than the other ones. In fact, the average maximum spectral efficiency more than doubles when the same cell radius is compared.

Finally, in terms of cell size, it is clear that for all the results presented here, the impact in terms of CDF, average maximum and minimum spectral efficiency is marginal which corroborates the argument that the system is not interference bounded.

Table 6. Maximum spectral efficiency per user in different cell sizes. The best value for each algorithm is are marked in bold type.

Parameters	Maximum Spectral Efficiency per User (in bps/Hz)				
	BPSO/GA	SPV–PSO		SPVG–PSO	
		w/o VNS	with VNS	w/o VNS	with VNS
$K = 20, R = 1$ Km	13.4015	29.3362	29.3362	29.3378	**29.3515**
$K = 20, R = 500$ m	12.9367	28.8354	28.9150	28.8825	**28.9445**
$K = 20, R = 250$ m	12.7704	28.7173	**28.8393**	28.7896	28.7380
$K = 20, R = 100$ m	13.1714	29.0950	29.0974	29.1499	**29.2260**

5.3. System Loading Impact on Performance

Figure 9 describes the empirical CDFs for different system loading scenarios: $K = 20$, $K = 40$ and $K = 60$. These values represent three system loading scenarios which represent low, medium and high user density. The main objective here is to determine whether the higher density of users in the system implies more interference among them.

It is evident that RA, BPSOand GA suffer an impact in performance as the system loading increases for the same cell size. However, that significant difference in performance is not observed for the other algorithms, meaning that they can handle user density better than the former three algorithms. Numerically, there is an 11 dB CDF shift from $K = 20$ users to $K = 40$ and a 7 dB shift from $K = 40$ to $K = 60$. Meanwhile, the performance difference between $K = 20$ and $K = 60$ is less than a 0.1 dB shift.

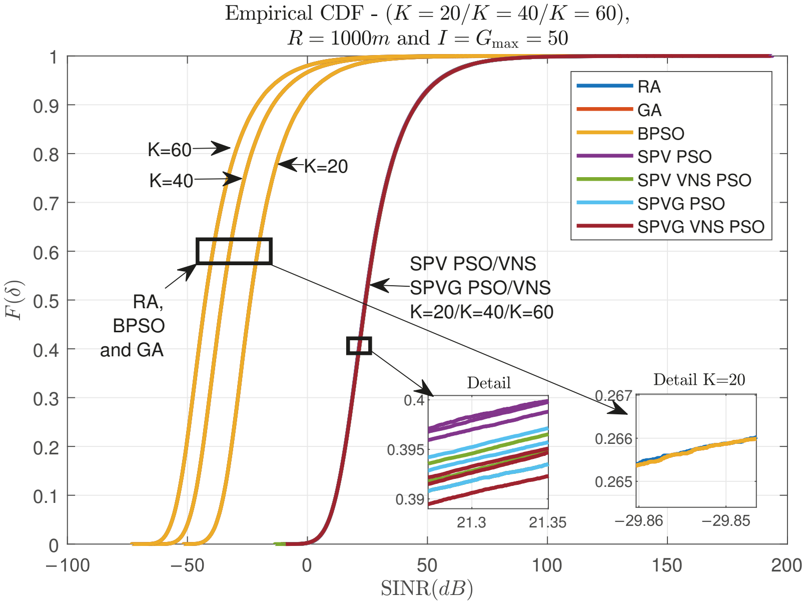


Figure 9. Empirical CDF for $K = 20$, $K = 40$ and $K = 60$ users in an R = 1000 m cell size scenario with the maximum number for iterations set to 50 for all algorithms.

6. Conclusions

This work presented six different heuristics to solve the pilot sequence allocation problem in Massive MIMO systems. We optimized the heuristics parameters to achieve the best performance in the discussed problem while still considering the computational time constraints inherent to the problem's nature.

We analyzed the impact of different scenario configurations on the performances of the algorithms. The cell size impact analysis offered clear conclusions that the system is not interference bounded and that the performance of SPV–PSO and all its derivations have far better performance in terms of system spectral efficiency, as well as average maximum and minimum spectral efficiency when compared to the RA, BPSO and GA solutions.

We further investigated the impact of system loading and found that RA, BPSO and GA solutions are directly affected by user density, while the other algorithms presented a more robust performance. We found that average minimum spectral efficiency may be increased in up to 1.7 million % when compared from RA, BPSO and GA solutions to a SPV–PSO-based one during the described investigations.

Finally, it is important to point out that SPV–PSO with $\omega = 0.1$, $v_{\max} = 3$, $c_1 = 5$ and $c = 1$ achieved the best results for the R = 1000 m and $K = 20$ users scenario.

7. Future Works

Future works include testing and comparing more different approaches to solve this problem such as: micro GAs, Differential Evolution, such as in [25], other versions of the PSO, as in [26,27], greed search algorithms, supervised learning with the Convolutional Neural network (CNN) and Multilayer Perceptron (MLP) as well as further investigating at which cell size or path-loss configurations the system becomes interference bounded.

Author Contributions: Conceptualization, L.D.H.S.; Investigation, E.A.M.; Methodology, D.S.S.; Project administration, L.D.H.S.; Software, E.A.M.; Supervision, D.S.S. and R.S.P.; Validation, R.P.B. and L.D.H.S.; Writing—original draft, E.A.M.; Writing—review & editing, R.P.B., R.S.P. and L.D.H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: github.com/evertonalex/utfpr-ppgi-pilotcontamination (accessed on 16 May 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cisco. Cisco Visual Networking Index: Forecast and Trends, 2018 2023 White Paper. Avaliable online: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html (accessed on 11 November 2021).
2. United Nations, Department of Economic and Social Affairs. World Population Prospects 2019. Available online: https://population.un.org/wpp/ (accessed on 11 November 2021).
3. Ericsson. Ericsson Mobility Report, Ericsson Mobility Report. Available online: https://www.ericsson.com/assets/local/reports-papers/mobility-report/documents/2019/emr-november-2019.pdf (accessed on 11 November 2021)
4. Pand, S.; Fitzek, F.H.P.; Lehmann, C.; Nophut, D.; Kiss, D.; Kovács, V.; Ákos, N.; Csovási, G.; Tóth, M.; Rajacsics, T. Joint Design of Communication and Control for Connected Cars in 5G Communication Systems. In Proceedings of the 2016 IEEE Globecom Workshops (GC Wkshps), Washington, DC, USA, 2016; pp. 1–7. [CrossRef]
5. Marzetta, T.L. Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Trans. Wirel. Commun.* **2010**, *9*, 3590–3600. [CrossRef]
6. Jin, S.; Li, M.; Huang, Y.; Du, Y.; Gao, X. Pilot scheduling schemes for multi-cell massive multiple-input–multiple-output transmission. *IET Commun.* **2015**, *9*, 689–700. [CrossRef]
7. Zhu, X.; Wang, Z.; Dai, L.; Qian, C. Smart Pilot Assignment for Massive MIMO. *IEEE Commun. Lett.* **2015**, *19*, 1644–1647. [CrossRef]

8. Ku, L.; Fan, J.; Deng, J. Low complexity pilot allocation in massive MIMO systems. In Proceedings of the 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN), Beijing, China, 4–6 June 2016. [CrossRef]
9. Teeti, M.A.; Wang, R.; Liu, Y.; Ni, Q. Pilot optimization in multicell massive MIMO. In Proceedings of the 2016 IEEE International Conference on Communication Systems (ICCS), Shenzhen, China, 14–16 December 2016. [CrossRef]
10. Zhou, Z.; Wang, D. Pilot scheduling based on water-filling algorithm in massive MIMO. In Proceedings of the 6th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 17–19 June 2016; pp. 89–92. [CrossRef]
11. Zhang, N.; Bai, Z.Q.; Zhang, B.; Su, Y.; Han, T.; Kwak, K. Genetic algorithm based pilot allocation scheme for massive MIMO system. In Proceedings of the 2016 IEEE International Conference on Communication Systems (ICCS), Shenzhen, China, 14–16 December 2016. [CrossRef]
12. Niu, G.; Mu, X.; Guo, X.; Zhang, J. Pilot Pollution Algorithm Based on Cell Sectorizatoin Elimination in Massive MIMO System. In Proceedings of the 7th IET International Conference on Wireless, Mobile & Multimedia Networks (ICWMMN 2017), Beijing, China, 17–20 November 2017. [CrossRef]
13. Makram Alkhaled, E.A.; Hamdi, K.A. Adaptive Pilot Allocation Algorithm for Pilot Contamination Mitigation in TDD Massive MIMO Systems. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017. [CrossRef]
14. Kim, K.; Lee, J.; Choi, J. Deep Learning Based Pilot Allocation Scheme (DL-PAS) for 5G Massive MIMO System. *IEEE Commun. Lett.* **2018**, *22*, 828–831. [CrossRef]
15. Gao, H.; Zhang, T.; Feng, C.; Wang, Y. Clustering Based Pilot Allocation Algorithm for Mitigating Pilot Contamination in Massive MIMO Systems. In Proceedings of the 2018 Asia-Pacific Microwave Conference (APMC), Kyoto, Japan, 6–9 November 2018; pp. 878–880. [CrossRef]
16. Khan, A.; Irfan, M.; Ullah, Y.; Ahmad, S.; Ullah, S.; Hayat, B. Pilot Contamination Mitigation for High and Low Interference Users in Multi-Cell Massive MIMO Systems. In Proceedings of the 2019 15th International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 2–3 December 2019; pp. 1–5. [CrossRef]
17. Luo, S.; Zhang, C.; Duan, Y.; Chen, J. Pilot Allocation Game: A Monte Carlo Tree Based Method. In Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 16–18 October 2019; pp. 277–282. [CrossRef]
18. Nie, X.; Zhao, F. Joint Pilot Allocation and Pilot Sequence Optimization in Massive MIMO Systems. *IEEE Access* **2020**, *8*, 60637–60644. [CrossRef]
19. Zhu, X.; Dai, L.; Wang, Z.; Wang, X. Weighted-Graph-Coloring-Based Pilot Decontamination for Multicell Massive MIMO Systems. *IEEE Trans. Veh. Technol.* **2017**, *66*, 2829–2834. [CrossRef]
20. Xu, S.; Zhang, H.; Tian, J.; Wu, D.; Yuan, D. Pilot Length Optimization for Spectral and Energy Efficient D2D Communications Underlay Massive MIMO Networks. In Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 5–8 March 2018; pp. 855–860.
21. Dey, A.; Pattanayak, P.; Gurjar, D.S. Arithmetic/Geometric Progression Based Pilot Allocation With Antenna Scheduling for Massive MIMO Cellular Systems. *IEEE Netw. Lett.* **2021**, *3*, 1–4. [CrossRef]
22. Parida, P.; Dhillon, H.S. Pilot Assignment Schemes for Cell-Free Massive MIMO Systems. *arXiv* **2021**, arXiv:2105.09505.
23. Kennedy, J.; Eberhart, R. A discrete binary version of the particle swarm algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 5, pp. 4104–4108. [CrossRef]
24. Holland, J.H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
25. Santucci, V.; Baioletti, M.; Milani, A. An algebraic framework for swarm and evolutionary algorithms in combinatorial optimization. *Swarm Evol. Comput.* **2020**, *55*, 100673. [CrossRef]
26. Chen, W.N.; Zhang, J.; Chung, H.S.H.; Zhong, W.L.; Wu, W.G.; Shi, Y.h. A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems. *IEEE Trans. Evol. Comput.* **2010**, *14*, 278–300. [CrossRef]
27. Moraglio, A.; Chio, C.; Togelius, J.; Poli, R. Geometric Particle Swarm Optimization. *J. Artif. Evol. Appl.* **2008**, *2008*, 143624. [CrossRef]

applied sciences

MDPI

Article

A Novel Whale Optimization Algorithm for the Design of Tuned Mass Dampers under Earthquake Excitations

Luis A. Lara-Valencia *, Daniel Caicedo and Yamile Valencia-Gonzalez

Facultad de Minas, Departamento de Ingeniería Civil, Universidad Nacional de Colombia, Sede Medellín,
Medellín 75267, Colombia; dcaicedod@unal.edu.co (D.C.); yvalenc0@unal.edu.co (Y.V.-G.)
* Correspondence: lualarava@unal.edu.co; Tel.: +57-310-407-7570

Featured Application: Adaptation of Whale Optimization Algorithm (WOA) for optimum design of TMDs. Seismic performance is assessed using real accelerograms and an irregular case study. Effective, fast, and reliable methodology comparing to others in the literature. Practical design recommendations are provided.

Abstract: This paper introduces a novel methodology for the optimum design of linear tuned mass dampers (TMDs) to improve the seismic safety of structures through a novel Whale Optimization Algorithm (WOA). The algorithm is aimed to reduce the maximum horizontal peak displacement of the structure, and the root mean square (RMS) response of displacements as well. Furthermore, four additional objective functions, derived from multiple weighted linear combinations of the two previously mentioned parameters, are also studied in order to obtain the most efficient TMD design configuration. The differential evolution method (DEM), whose effectiveness has been previously demonstrated for TMD applications, and an exhaustive search (ES) process, with precision to two decimal positions, are used to compare and validate the results computed through WOA. Then, the proposed methodology is applied to a 32-story case-study derived from an actual building, and multiple ground acceleration time histories are considered to assess its seismic performance in the linear-elastic range. The numerical results show that the proposed methodology based on WOA is effective in finding the optimal TMD design configuration under earthquake loads. Finally, practical design recommendations are provided for TMDs, and the robustness of the optimization is demonstrated.

Keywords: tuned mass dampers; whale optimization algorithm; differential evolution method; optimal design; earthquake loads

Citation: Lara-Valencia, L.A.; Caicedo, D.; Valencia-Gonzalez, Y. A Novel Whale Optimization Algorithm for the Design of Tuned Mass Dampers under Earthquake Excitations. *Appl. Sci.* **2021**, *11*, 6172. https://doi.org/10.3390/app11136172

Academic Editor: Eui-Nam Huh

Received: 28 May 2021
Accepted: 28 June 2021
Published: 2 July 2021

1. Introduction

Structural control systems have turned into a standard technology to improve the dynamic response of civil engineering structures subjected to dynamic actions, such as wind forces or earthquake loads [1]. These control systems can be classified into four major groups: passive, active, hybrid, and semiactive controllers. Passive systems are widely accepted by the engineering community because of their mechanical simplicity, low power requirements, and controllable force capacity [2]. Among passive systems, one of the most commonly used and tested devices throughout the years has been the tuned mass damper (TMD). This system consists of attaching an additional mass linked to the main structure using a spring and a viscous damper, which is optimally tuned to one of the fundamental vibration frequencies of the system in order to transfer energy among the vibrating modes by making the structure more flexible [3]. The original TMD formulation was first proposed by Frahm in [4] as a vibration absorber with no damping to control periodic resonance vibrations. Subsequently, Ormondroyd and Den Hartog [5] developed the theory of vibration absorbers, including viscous damping to the system to be effective under different frequencies of random vibrations. Conventional tuning methodologies have been proposed thereafter, considering the harmonic loads, and random stationary and nonstationary white noise processes for single degree of freedom systems [6–9].

The first full-scale implementations of TMDs were aimed to control dynamic displacements caused by wind-induced vibrations. In that sense, TMDs have been deployed in several structures around the world, including the Centre Point Tower in Sydney, the Citicorp Center in New York, the Chiba Port Tower in Japan, the CN Tower in Canada, and, more recently, the Taipei 101 tower in Taiwan [10,11]. Several variations based on the conventional TMD formulation have been studied and applied thereafter, including pendulum-tuned mass dampers (PTMD) [12–14], tuned liquid column dampers (TLCD) [15–17], bidirectional TMDs [18] and hybrid, semi-active, and active TMDs [19–21]. However, some early investigations concluded that TMDs were not effective in reducing the response of buildings subjected to seismic excitation [22,23], which was due to two principal reasons. First, the limitations associated with the amount of mass added to the structural system, and, more importantly, that conventional TMDs can suppress effectively a single vibration mode. Instead, wide-band multi-modal damping can be achieved by complex systems, such as nonlinear TMDs [24,25], multiple positioned TMDs [26], or the amplification of the inertial mass in conventional TMDs using inerter devices (TMDIs) [27–29].

In spite of this, since the early 2000s, some researchers have proved the effectiveness of numerical iterative methods in the tuning of linear mass dampers for seismic applications. Classic techniques have been used for that purpose, such as minimax optimization [30], response surface methodology [31], and nonlinear gradient-based optimization [32]. Moreover, conventional population-based metaheuristics and evolutionary algorithms have been used to improve the computational efficiency in the tuning procedure; among them, particle swarm optimization [33,34]. harmony search [35–40], ant colony [41,42], flower pollination algorithm [43–45], bat algorithm [46], cuckoo search [47], genetic algorithms [48–50], and, more recently, chaotic optimization [51]. Furthermore, various of these investigations have included the effects of dynamic soil–structure interaction (SSI) [52–55]. Even though the tuning of linear TMDs on single degree of freedom systems is a very well addressed problem in the literature, most of these works were based on the assumption that closed-form expressions, like those presented in [6–9], are not valid for multi-degree of freedom systems subjected to actual seismic loads. On the contrary, the usage of metaheuristic techniques or evolutionary algorithms allows the best-fit design variables for linear TMDs to be determined, using actual accelerograms as input excitations.

In that sense, this study introduces a novel methodology for the optimal design of passive TMDs located at upper levels of high-rise buildings to improve the seismic safety of structures based on the Whale Optimization Algorithm (WOA) [56]. The algorithm is modified to reduce the maximum horizontal peak displacement of the structure, and the root mean square (RMS) response of displacements as well. Moreover, four additional objective functions, derived from multiple weighted linear combinations of these parameters, are also studied in order to obtain the most efficient TMD design configuration. The results from the WOA optimization are compared with the differential evolution method (DEM) [57], whose effectiveness has been demonstrated extensively for TMDs and TMDIs seismic applications in previous works [58,59], and an exhaustive search (ES) process with precision to two decimal positions. Then, the proposed methodology is applied to a 32-story case-study derived from an actual building structure, and various accelerograms of recorder earthquakes are considered to assess its seismic performance in the linear-elastic range. The results of the study show a large enhancement in the dynamic response of the building controlled by the WOA designed TMDs, achieving reductions in the maximum floor displacements of up to 43%. Finally, an attempt to establish a single set of design parameters for TMDs based on the methodology proposed by Fallah and Zamiri [60] is implemented to verify the robustness of the optimization.

2. Equations of Motion of *n*-DOF Building Structures Controlled via TMDs for Seismic Applications

The equations of motion that govern the behavior of the *n*-DOF structural system equipped with a TMD located at the top floor of the building are deduced from the scheme depicted in Figure 1. The displacement at each story-level is represented by x_i, while the stiffness, damping, and mass coefficients of each story-level of the building are defined as k_i, c_i, and m_i, respectively. Besides, the TMD action is introduced as an extra DOF represented by x_d, while the stiffness, damping, and mass parameters are identified as k_d, c_d, and m_d, respectively.

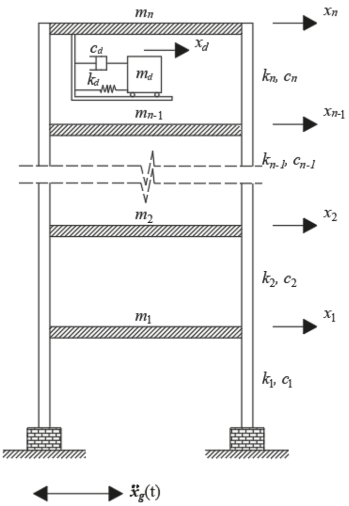


Figure 1. *n*-DOF system controlled via TMDs for seismic applications.

Subsequently, the structural system controlled via TMD can be modeled as a set of *n* + 1 degrees of freedom, whose equation of motion can be expressed as:

$$\mathbf{M}\ddot{\mathbf{x}}(t)+\mathbf{C}\dot{\mathbf{x}}(t)+\mathbf{K}\mathbf{x}(t)=-\mathbf{M}1\ddot{x}_g(t) \tag{1}$$

where, $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$, and $\ddot{\mathbf{x}}(t)$ are the *n* + 1-order vectors of displacement, velocity, and acceleration of the system; **1** is a unit vector, and $\ddot{x}_g(t)$ denotes the ground acceleration over time. On the other hand, the matrices **M**, **C,** and **K** represent the mass, damping, and stiffness matrices of the system, which can be expressed as:

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & m_2 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & m_3 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & m_{n-1} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & m_n & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 0 & m_d \end{bmatrix} \tag{2}$$

$$\mathbf{C} = \begin{bmatrix} c_1+c_2 & -c_2 & 0 & 0 & \cdots & \cdots & 0 \\ -c_2 & c_2+c_3 & -c_3 & 0 & \cdots & \cdots & 0 \\ 0 & -c_3 & c_3+c_4 & -c_4 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & -c_{n-1} & c_{n-1}+c_n & -c_n & 0 \\ \vdots & \vdots & \vdots & 0 & -c_n & c_n+c_d & -c_d \\ 0 & 0 & \cdots & 0 & 0 & -c_d & c_d \end{bmatrix} \tag{3}$$

$$\mathbf{K} = \begin{bmatrix} k_1+k_2 & -k_2 & 0 & 0 & \cdots & \cdots & 0 \\ -k_2 & k_2+k_3 & -k_3 & 0 & \cdots & \cdots & 0 \\ 0 & -k_3 & k_3+k_4 & -k_4 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & -k_{n-1} & k_{n-1}+k_n & -k_n & 0 \\ \vdots & \vdots & \vdots & 0 & -k_n & k_n+k_d & -k_d \\ 0 & 0 & \cdots & 0 & 0 & -k_d & k_d \end{bmatrix} \tag{4}$$

It should be noted that matrices (2) to (4) are idealized as tridiagonal stiffness matrices typical of shear frame structures. Nevertheless, the main purpose of these mathematical expressions is to illustrate the inclusion of the TMD effect on a linear structural system with *n* horizontal degrees of freedom, and, therefore, it can be extended to other structural models such as 2-dimensional frames derived from actual building numerical models with a static condensation applied to all other vertical and rotational degrees of freedom. Now, the equation of motion of the system is modified to a space–state representation to determine the dynamic response of the linear system. Hence, the state vector $\mathbf{z}(t)$ is introduced as:

$$\mathbf{z}(t) = \begin{Bmatrix} \mathbf{x}(t) \\ \dot{\mathbf{x}}(t) \end{Bmatrix} \tag{5}$$

The above-mentioned vector involves the displacements and velocities of the controlled system. By deploying Equation (1), the state–space representation of the structural system controlled by TMD can be written as:

$$\begin{Bmatrix} \dot{\mathbf{x}}(t) \\ \ddot{\mathbf{x}}(t) \end{Bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \dot{\mathbf{x}}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{M}^{-1} \end{bmatrix} \cdot -\mathbf{M}1\ddot{x}_g(t) \tag{6}$$

where **0** and **I** denote a zero matrix and an identity matrix, respectively. Subsequently, Equation (6) is rewritten as:

$$\dot{\mathbf{z}}(t) = \mathbf{A}z(t) + \mathbf{B} \cdot -\mathbf{M}1\ddot{x}_g(t) \tag{7}$$

where **A** represents the transition state matrix, and **B** the location adjustment matrix of the external excitation in the structural system:

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \tag{8}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{M}^{-1} \end{bmatrix} \tag{9}$$

Taking inspiration from the tuning process of TMDs on single degree of freedom systems, and to simplify the optimization process, the k_d and c_d parameters can be rewritten as:

$$k_d = \omega_s{}^2 f^2 m_d \tag{10}$$

$$c_d = 2\zeta_d f \omega_s m_d \tag{11}$$

with ω_s as the natural frequency of the structural system and f and ζ_d as the dimensionless frequency and damping ratios to be optimized. The reference value ω_s will be assumed in this investigation as the corresponding frequency to the first vibration mode of the structure. Nevertheless, this is just an assumption to reduce significantly the search domain of the variables k_d and c_d since the optimization problem is focused on the tuning problem for TMDs on multi-degree-of-freedom systems.

3. Whale Optimization Algorithm

The Whale Optimization Algorithm (WOA) is a swarm-based metaheuristic recently proposed by Mirjalili and Lewis in [56], who developed the algorithm by taking inspiration from the bubble-net hunting strategy typically employed by humpback whales. In the last few years, other researchers have applied multiple modifications to the algorithm in order to solve various engineering optimization problems. For instance, Kaveh and Ghazaan [61] studied the sizing optimization of skeletal structures, Chen et al. [62] solved bar truss design and I-beam design problems, and Azizi et al. [63] optimized a fuzzy controller for seismically excited nonlinear structures.

WOA strategy first defines a population of whales and evaluates a set of random solutions according to both the position of each whale and a certain objective function. By comparing the primary random solutions, WOA selects an initial best search agent. In each iteration, whales update their position with respect to either a randomly chosen search agent or the best solution obtained so far. Then, the new positions are calculated according to Equations (12) and (13):

$$\Delta = |\Gamma \cdot \mathbf{X}^*(w) - \mathbf{X}(w)| \tag{12}$$

$$\mathbf{X}(w+1) = \mathbf{X}^*(w) - \Psi \cdot \Delta \tag{13}$$

where w is the current whale generation, and $\mathbf{X}^*$ is the updated position vector of the best solution. In addition, the coefficient vectors Ψ and Γ are calculated as follows:

$$\Psi = 2a \cdot \mathbf{r} - a \tag{14}$$

$$\Gamma = 2\mathbf{r} \tag{15}$$

with $\mathbf{r}$ as a random value in the domain (0, 1). Besides, a denotes a sequentially decreasing number from 2 to 0 as whales generation w increases, to provide exploration and exploitation.

According to Mirjalili and Lewis [56], a random search agent is chosen when $|\Psi| \geq 1$, while the best solution is selected when $|\Psi| < 1$ for updating the position of the search agents. The bubble-net feeding behavior of humpback whales serves as an inspiration to update the position of the search agents. It is assumed that there is the same probability of selecting between either a shrinking encircling mechanism or a spiral model, to improve the position of the search agents during the optimization procedure and get closer to the optimal solution. This concept is mathematically formulated as:

$$\mathbf{X}(w+1) = \begin{cases} \mathbf{X}^*(w) - \Psi \cdot \Delta & \text{if } p < 0.5 \\ \Delta' \cdot e^{bl} \cdot \cos(2\pi l) + \mathbf{X}^*(w) & \text{if } p \geq 0.5 \end{cases} \tag{16}$$

where b is a constant that defines the shape of the logarithmic spiral; l is a random number in the space $(-1, 1)$; p is a random probability in (0, 1). In the adaptation of the algorithm, the constant b was set to 1 following the recommendations provided in [61–63]. On the other hand, the l and p parameters are chosen randomly in their respective domain for every generation w, in order to avoid convergence at local minimums and to improve the efficiency of the algorithm. Finally, Δ' defines the distance of the ith whale to the best solution obtained so far, which could be calculated as:

$$\Delta' = |\mathbf{X}^*(w) - \mathbf{X}(w)| \tag{17}$$

Six objective functions are proposed in the optimization process to diminish the dynamic response of the structure when it is subjected to seismic excitations. These functions are associated with the reduction of the maximum horizontal peak of displacement of the structure, and the reduction of the root mean square (RMS) values of displacements. Certainly, there are other critical parameters to be analyzed in a robust structural design, such as inter-story drifts and peak floor accelerations, which are closely related to structural damage. Nevertheless, the criteria for selecting the horizontal peak displacements and RMS values of displacements as critical parameters in the tuning procedure responds to two principal reasons. Firstly, the reduction of absolute displacements in the objective functions contributes directly to diminishing the inter-story drift values and, more importantly, the main objective of this research is to propose a novel tuning methodology based on WOA, as well as proving its efficiency over other optimization techniques [30–55]. Moreover, the usage of multiple functions aims to determine which of these objective functions is more efficient, based on the evaluation and comparison on the decrease in the response of both previously mentioned parameters.

In that sense, the objective functions OA1 and OA2 correspond to the reduction of the maximum horizontal peak displacement in the entire structure, and the decrease in the RMS values of displacements, respectively:

$$\mathrm{OA1} = \min(\max(|x_n|)) \quad \text{for } x_n \ = [x_1,\ x_2,\ \ldots,\ x_n] \tag{18}$$

$$\mathrm{OA2} = \min(\max(RMS(x_n))) \quad \text{for } x_n \ = [x_1,\ x_2,\ \ldots,\ x_n] \tag{19}$$

Regarding the remaining four objective functions, these are defined as a variable weighted linear combination of the relationships between the above-mentioned parameters in Equations (18) and (19). These functions are defined as J_1, J_2, J_3, and J_4, and are described through expression (20):

$$\mathrm{J}_i = \min\left((i * 0.2)\frac{\max(|x_n|)}{\max\ (|x_n^*|)} + (1 - (i * 0.2))\frac{\max(\mathrm{RMS}(x_n))}{\max\ (\mathrm{RMS}(x_n^*))}\right) \text{for } i = 1:4 \text{ and } x_n \ = [x_1,\ x_2,\ \ldots,\ x_n] \tag{20}$$

where x_n^*. denotes the uncontrolled response of the structure at the n degree of freedom of the structure, introduced to normalize the values of J_i function. Different from OA1 and OA2 functions, in which absolute values can be used to analyze the numerical results computed from the optimization process, the function J_i combines horizontal peak displacements and RMS values of displacements which are commonly different in magnitude. Hence, both values must be normalized to be properly weighted and combined thereafter.

The TMD tuning procedure in multi-degree of freedom systems is a two-dimensional optimization problem focused on finding the optimal k_d and c_d values for a fixed mass ratio (μ). As mentioned in Section 2, the optimization process is simplified by the inclusion of Equations (10) and (11). Hence, WOA is programed to find the optimal f and ζ_d values in the domain:

$$0.50 \le f \le 2.00 \tag{21}$$

$$0 \le \zeta_d \le 0.50 \tag{22}$$

The bounds in the search domain were defined according to practical recommendations that were taken from [30–55]. The flowchart in Figure 2 summarizes the WOA procedure.

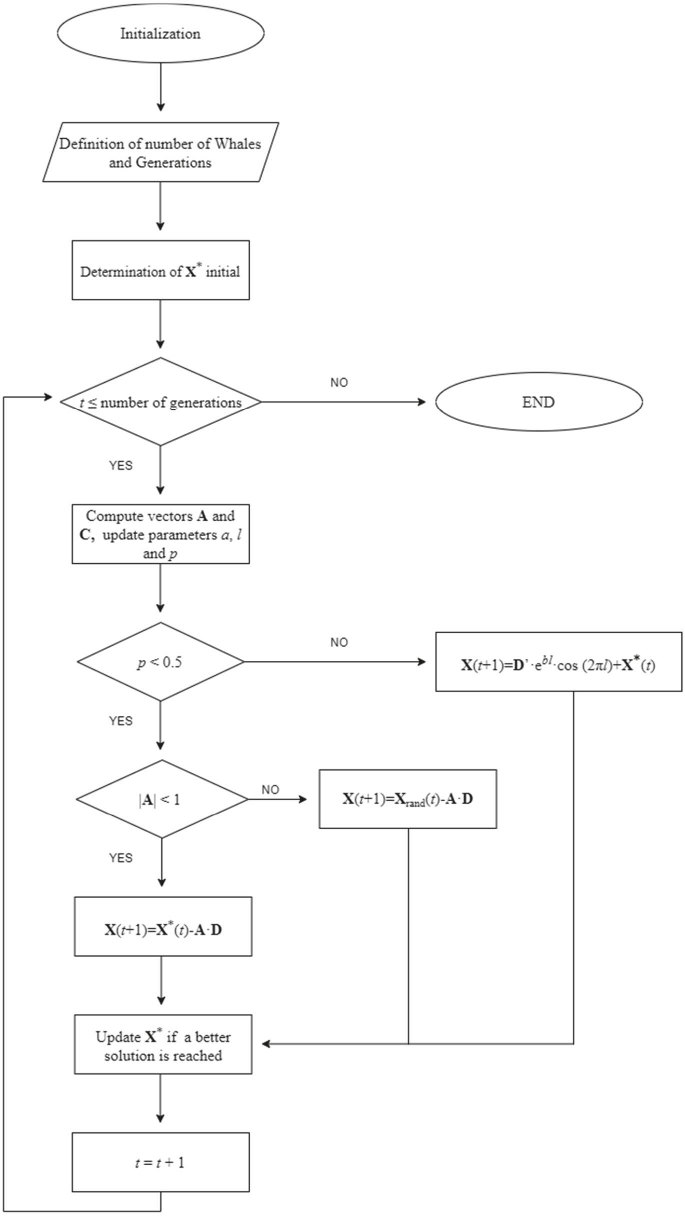

Initialization
Definition of number of Whales and Generations
Determination of $\mathbf{X}^*$ initial
$t \leq$ number of generations
NO
END
YES
Compute vectors **A** and **C**, update parameters a, l and p
$p < 0.5$
NO
$\mathbf{X}(t+1)=\mathbf{D}'\cdot e^{bl}\cdot\cos(2\pi l)+\mathbf{X}^*(t)$
YES
$|\mathbf{A}| < 1$
NO
$\mathbf{X}(t+1)=\mathbf{X}_{rand}(t)-\mathbf{A}\cdot\mathbf{D}$
YES
$\mathbf{X}(t+1)=\mathbf{X}^*(t)-\mathbf{A}\cdot\mathbf{D}$
Update $\mathbf{X}^*$ if a better solution is reached
$t = t + 1$

Figure 2. Description of the WOA procedure.

4. Numerical Example

In this section, the proposed methodology is applied to a 32-story case-study derived from an actual building in Medellin city, Colombia. The structure has a total elevation of 97 m, and the lateral force-resisting system consists of resistant moment frames. Besides, the structure has an irregular L-shape configuration in-plan, which makes it more vulnerable to seismic damage. The resulting mass, and stiffness matrix are 32 × 32 size, obtained by assuming in-plane infinitely rigid floor diaphragms and applying static condensation on the remaining vertical and rotational degrees of freedom. A complete characterization of the case-study can be found in the work presented by Caicedo et al. in [59]. Once the design parameters computed through WOA have been compared with the differential evolution method (DEM) and an exhaustive search (ES), the structural system is subjected to the action of the ground acceleration records used in the optimization process to assess its seismic performance by considering only the linear-elastic behavior of the structure. Then, the methodology proposed by Fallah and Zamiri [60] is adopted, in which, through a weighting procedure, a unique set of tuning parameters for a fixed μ value is proposed to verify the robustness of the design under different records than those used in the optimization process.

4.1. Benchmark Records

Four widely known accelerograms in the literature of recorded earthquakes were downloaded from the Pacific Earthquake Engineering Research (PEER) Centre database [64] to simulate the seismic action in the optimization process. The four records are labeled and described in Table 1.

Table 1. Characterization of the benchmark records.

Record	Event Name	Station	Component	PGA [g]	Duration [s]
1	El Centro	El Centro	S00E	0.35	53.73
2	Kobe	Takatori	000 (CUE)	0.61	30.03
3	Loma Prieta	Capitolia	90 DEG	0.40	39.98
4	Northridge	Rinaldi	S49W	0.84	14.97

The four accelerograms listed in Table 1 correspond to historical ground motions from different locations and, certainly, all four records present very different dynamic characteristics. Moreover, it is impossible to know a priori future ground motions that will affect the structural system. Therefore, it should be clarified that the intention of this research is not to reproduce a realistic tuning process for TMDs subjected to earthquake loads; on the contrary, and as with other recently proposed methodologies [44–46,51,58,59], this paper’s purpose is to prove the efficiency of the WOA to find the best-fit design variables for TMDs to improve the dynamic response of multi-story structures using actual accelerograms as input excitations, and examine extensively the dynamic behavior of the structure under the action of such accelerograms.

4.2. Calibration of the Algorithm

The main challenge of adapting the WOA methodology is to find a balance between the algorithm performance and the processing time. In that sense, multiple numerical simulations were carried out for the case study, using the objective function J_1 and the Kobe excitation. Alternatives of 10, 30, 50, 100, 200, and 500 whales combined with 5, 10, 30, and 50 iterations were considered. Figure 3 depicts the performance index (PI), defined as the ratio between the maximum controlled and uncontrolled response of the system, and the processing time for every simulation.

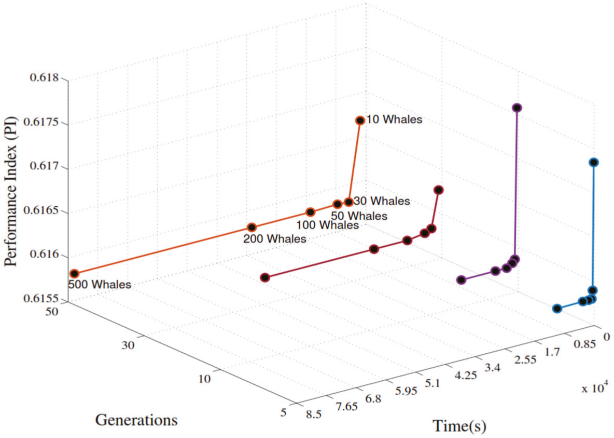


Figure 3. Calibration process of the optimization algorithm.

Furthermore, Table 2 reports the design parameters calculated for each alternative, as a complement to the information detailed in Figure 3.

Table 2. Complementary information on the calibration process.

Generations	Number of Whales	f	ζ_d	PI	Time (s)
5	10	0.98864896	0.22564242	0.617347181	169.58
	30	1.00616516	0.17197331	0.615902907	504.07
	50	1.01274866	0.17694892	0.615874230	840.77
	100	1.00587946	0.18405395	0.615803508	1679.90
	200	1.00662822	0.18524315	0.615805754	3287.17
	500	1.00660307	0.18278162	0.615801297	10628.89
10	10	1.00007356	0.23479706	0.617584632	336.71
	30	1.00778482	0.17337128	0.615877755	982.16
	50	1.00759291	0.17604717	0.615840363	1712.72
	100	1.00688244	0.18293491	0.615801209	3295.85
	200	1.00677897	0.18340897	0.615801449	6502.30
	500	1.00691211	0.18258605	0.615801269	16239.38
30	10	1.00318603	0.20829184	0.616279278	987.30
	30	1.00560264	0.19178801	0.615864246	3040.44
	50	1.00602357	0.18895977	0.615830943	4927.09
	100	1.00689775	0.18258902	0.615801266	9913.04
	200	1.00682148	0.18313549	0.615801265	19411.94
	500	1.00687972	0.18277428	0.615801211	50530.47
50	10	1.00153517	0.21820159	0.61668986	1614.58
	30	1.00692485	0.18242744	0.615801361	4874.36
	50	1.00640186	0.18620664	0.615810253	8089.97
	100	1.00690462	0.18253989	0.615801298	15816.19
	200	1.00685867	0.18287912	0.615801204	32551.09
	500	1.00685883	0.18287700	0.615801204	83204.51

Based on the results presented in Figure 3 and Table 2, it is possible to state that, in terms of performance and time, the most attractive optimization alternatives are those that use five generations and 50 whales (Alternative A), and 10 generations and 30 whales (Alternative B). Comparing the performance of alternatives A and B with the one that presents the best PI reduction, which corresponds to 50 generations and 500 whales (alternative C), diminutions in the processing time of 98.98% and 98.82%, respectively, are obtained.

In the same way, the differences in the PI values of alternatives A and B compared with alternative C are 0.0119% and 0.0124%, respectively. Considering these data and taking into account the stability of the methodology by employing a larger number of generations, alternative B is adopted in this work for the determination of optimal design parameters of TMDs.

4.3. Optimization Results

Table 3 presents the tuning parameters and the processing time for every iteration found for objective functions OA1 and OA2, and mass ratios μ = 0.02 and μ = 0.05, by applying the WOA methodology proposed herein. Additionally, a comparison is made with the DEM and an ES process with precision to two decimal positions. It should be noted that DEM has proved to be effective in solving the tuning problem of TMDs and TMDIs under actual earthquake excitations [58,59]; thus, it was selected to validate the WOA optimization results. Moreover, the work of Caicedo et al. [58] demonstrated the feasibility of DEM over other conventional tuning methodologies considering harmonic loads [6], white noise processes [7,8], and frequency domain analysis [9]. Similarly, Table 4 shows ζ_d and f parameters derived from J_1, J_2, J_3, and J_4 approaches.

As expected, the results computed by WOA and the other two comparison methodologies show correspondence. From Tables 3 and 4, it can be observed that some of the design variables approach the limits established previously in Equations (21) and (22). Such results are not realistic from a practical point of view, but they may be attributed to the impulsive nature of some of the seismic records used in the optimization process. However, in all cases the results are in excellent agreement with those computed through the DEM and the ES process, showing small differences from the third decimal position, which has no influence on the global response of the structural system. Furthermore, the numerical results reported by other works in which actual seismic records were used as excitation inputs [46,58,59,65] exhibited a similar trend, since some of the optimal values are also close to the bound limits defined for the optimization process. On the other hand, WOA exhibits notable advantages against ES and DEM, among them, less computational cost avoiding operations, like mutation or crossover, and reductions up to 45% in the processing time, approximately (see Tables 3 and 4). Figure 4 illustrates the dispersion of optimal design values derived from the three optimization methodologies and the six objective functions used in this investigation.

Table 3. TMD design parameters optimized by objective function OA1 and OA2.

Earthquake Ground Motion	μ = 0.02									μ = 0.05								
	DEM			ES			WOA			DEM			ES			WOA		
	ζ_d	f	Time (s)	ζ_d	f	Time (s)	ζ_d	f	Time (s)	ζ_d	f	Time (s)	ζ_d	f	Time (s)	ζ_d	f	Time (s)
OA1																		
El centro	0.48	2.00	3821.31	0.50	2.00	18934.41	0.50	2.00	1721.31	0.40	2.00	3736.23	0.39	2.00	18647.96	0.39	2.00	1660.55
Kobe	0.01	0.95	2135.75	0.00	0.94	10582.55	0.01	0.94	962.05	0.36	1.40	2088.20	0.35	1.38	10422.45	0.40	1.31	928.09
Loma Prieta	0.03	1.09	2843.40	0.01	0.96	14088.92	0.00	0.95	1280.81	0.20	0.89	2780.10	0.16	0.89	13875.78	0.15	0.89	1235.60
Northridge	0.50	2.00	1064.68	0.50	2.00	5275.42	0.50	2.00	479.58	0.00	1.21	1040.97	0.50	2.00	5195.61	0.50	2.00	462.65
OA2																		
El centro	0.21	0.90	3898.77	0.18	0.91	19003.27	0.18	0.91	1677.40	0.33	0.98	3816.45	0.34	0.98	18792.44	0.35	0.97	1699.75
Kobe	0.07	1.01	2179.04	0.07	1.00	10621.03	0.07	1.00	937.51	0.18	1.01	2133.04	0.17	1.01	10503.20	0.23	1.01	950.00
Loma Prieta	0.13	0.93	2901.04	0.10	0.90	14140.16	0.10	0.90	1248.14	0.16	0.86	2839.78	0.14	0.86	13983.28	0.16	0.85	1264.77
Northridge	0.14	0.93	1086.26	0.12	0.92	5294.60	0.11	0.92	467.35	0.22	0.84	1063.32	0.22	0.83	5235.86	0.22	0.84	473.58

Table 4. TMD design parameters optimized by objective function J_1, J_2, J_3, and J_4.

Earthquake Ground Motion	μ = 0.02									μ = 0.05								
	DEM			ES			WOA			DEM			ES			WOA		
	ζ_d	f	Time (s)	ζ_d	f	Time (s)	ζ_d	f	Time (s)	ζ_d	f	Time (s)	ζ_d	f	Time (s)	ζ_d	f	Time (s)
J_1																		
El centro	0.22	0.94	3901.19	0.21	0.93	19252.88	0.20	0.93	1757.29	0.35	1.03	3936.75	0.35	1.02	21074.23	0.35	1.02	1771.54
Kobe	0.07	1.01	2180.40	0.07	1.00	10760.54	0.08	1.00	982.16	0.18	1.01	2200.27	0.18	1.01	11778.50	0.17	1.01	990.12
Loma Prieta	0.11	0.94	2902.84	0.09	0.91	14325.89	0.10	0.91	1307.58	0.18	0.85	2929.30	0.16	0.89	15681.14	0.17	0.87	1318.19
Northridge	0.14	0.93	1086.93	0.12	0.93	5364.15	0.12	0.93	489.61	0.22	0.84	1096.84	0.22	0.84	5871.60	0.23	0.84	493.58
J_2																		
El centro	0.33	0.98	3898.77	0.31	0.98	20108.75	0.32	0.97	1715.43	0.37	1.06	3585.03	0.36	1.08	17755.27	0.31	1.10	1607.13
Kobe	0.10	1.04	2179.04	0.07	1.00	11238.90	0.08	0.99	958.76	0.22	1.03	2003.69	0.20	1.01	9923.52	0.21	1.01	898.24
Loma Prieta	0.06	0.95	2901.04	0.08	0.93	14962.74	0.08	0.93	1276.43	0.18	0.85	2667.59	0.16	0.89	13211.53	0.17	0.89	1195.85
Northridge	0.10	0.93	1086.26	0.12	0.93	5602.61	0.12	0.93	477.94	0.23	0.86	998.84	0.22	0.85	4946.89	0.23	0.85	447.77
J_3																		
El centro	0.50	1.22	3821.31	0.50	1.27	21056.73	0.50	1.22	1792.82	0.41	1.17	3727.83	0.42	1.19	18425.86	0.41	1.17	1653.28
Kobe	0.10	1.04	2135.75	0.06	0.99	11768.72	0.05	0.99	1002.02	0.31	1.03	2083.50	0.27	1.01	10298.31	0.31	1.02	924.03
Loma Prieta	0.06	0.95	2843.40	0.06	0.95	15668.12	0.05	0.95	1334.02	0.20	0.89	2773.84	0.16	0.89	13710.51	0.20	0.89	1230.19
Northridge	0.10	0.93	1064.68	0.12	0.94	5866.73	0.13	0.92	499.51	0.20	0.89	1038.63	0.23	0.87	5133.73	0.22	0.88	460.63
J_4																		
El centro	0.48	1.99	3898.77	0.50	2.00	17853.50	0.50	2.00	1622.96	0.41	1.98	3753.69	0.42	2.00	18590.50	0.42	2.00	1669.92
Kobe	0.01	0.95	2179.04	0.03	0.98	9978.42	0.08	0.97	907.08	0.48	1.19	2097.96	0.48	1.18	10390.33	0.50	1.15	933.33
Loma Prieta	0.02	0.95	2901.04	0.03	0.99	13284.63	0.04	0.96	1207.63	0.20	0.89	2793.09	0.16	0.89	13833.02	0.21	0.89	1242.58
Northridge	0.13	0.99	1086.26	0.11	0.97	4974.26	0.11	0.98	452.18	0.22	0.94	1045.84	0.23	0.93	5179.60	0.22	0.95	465.27

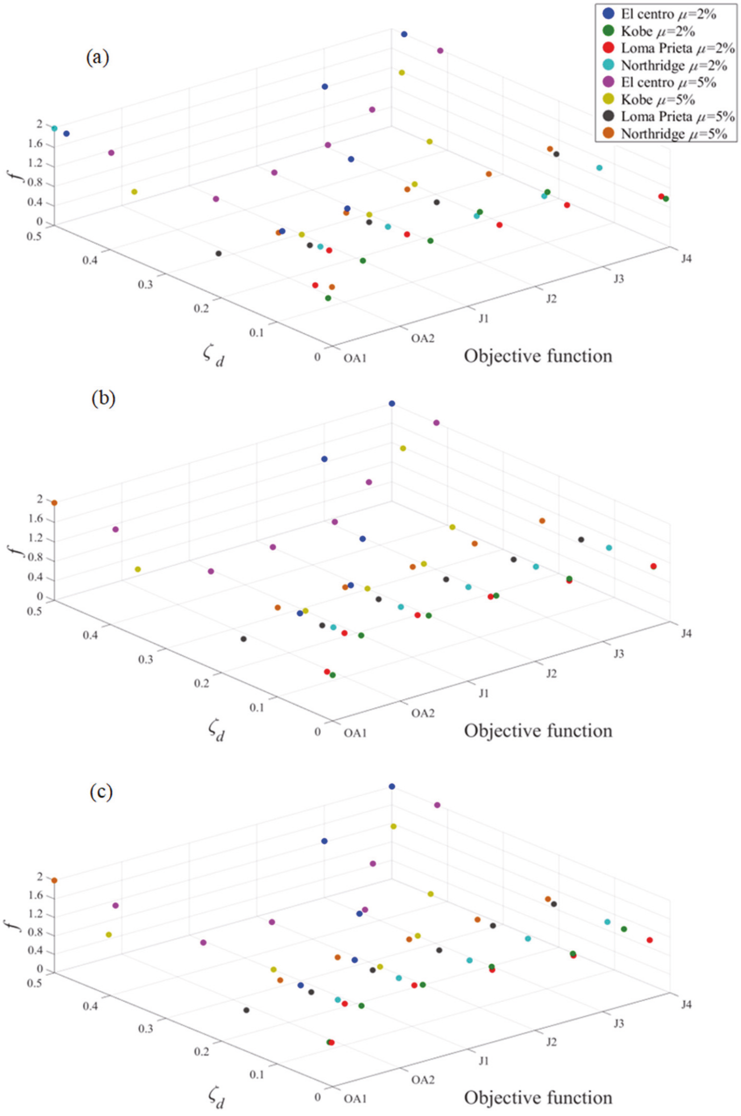
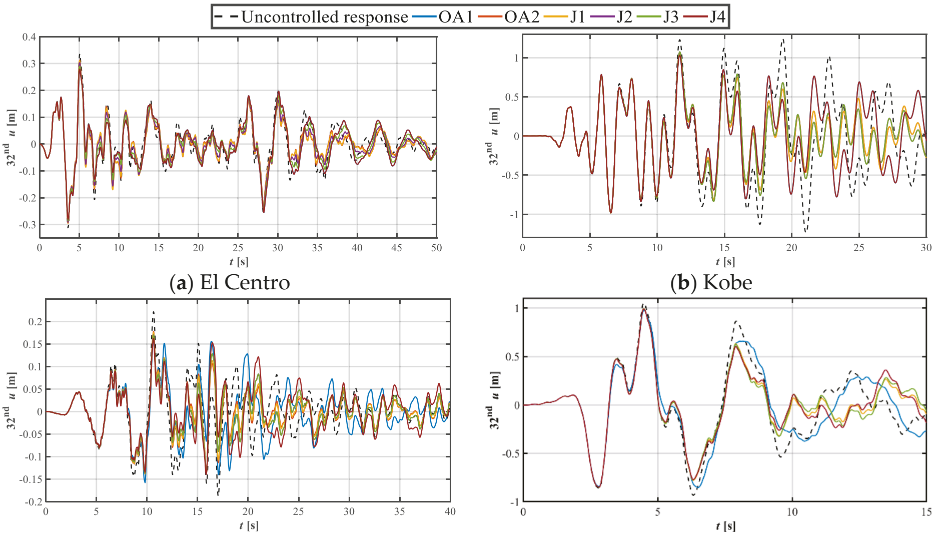


Figure 4. Scatter graphs for TMD design parameters: (**a**) DE; (**b**) ES; (**c**) WOA.

4.4. Seismic Performance

Now, the seismic performance of the building is evaluated under the action of the four seismic records used in the optimization process. It is worth noting that the scope of this investigation is limited to the linear behavior of the structural system. Thus, the numerical model does not take into account any type of damage or yielding process affecting the structure. Furthermore, the computed displacements are presumable within the elastic behavior of the system.

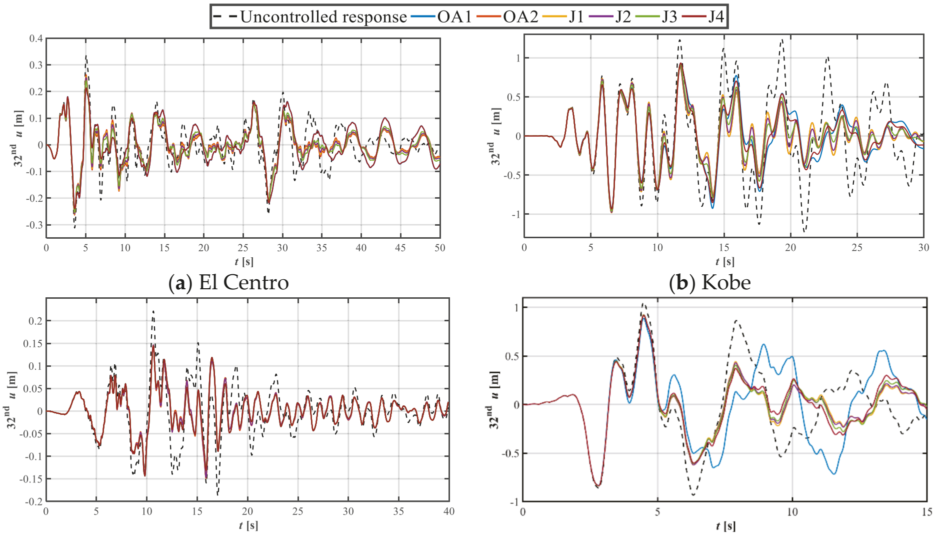
Figures 5 and 6 show the history of displacements at the 32nd story of the building, for μ = 0.02 and 0.05, respectively. Even though both TMD options (μ = 0.02, and 0.05) can effectively reduce the dynamic response of displacements, it is visibly clear that TMDs with μ = 0.05 reach greater reductions in lateral displacements at the upper level of the structural system, regardless of the objective function.



(**a**) El Centro (**b**) Kobe

(**c**) Loma Prieta (**d**) Northridge

Figure 5. Displacement of the 32nd story of the structure equipped with TMDs with μ = 0.02.

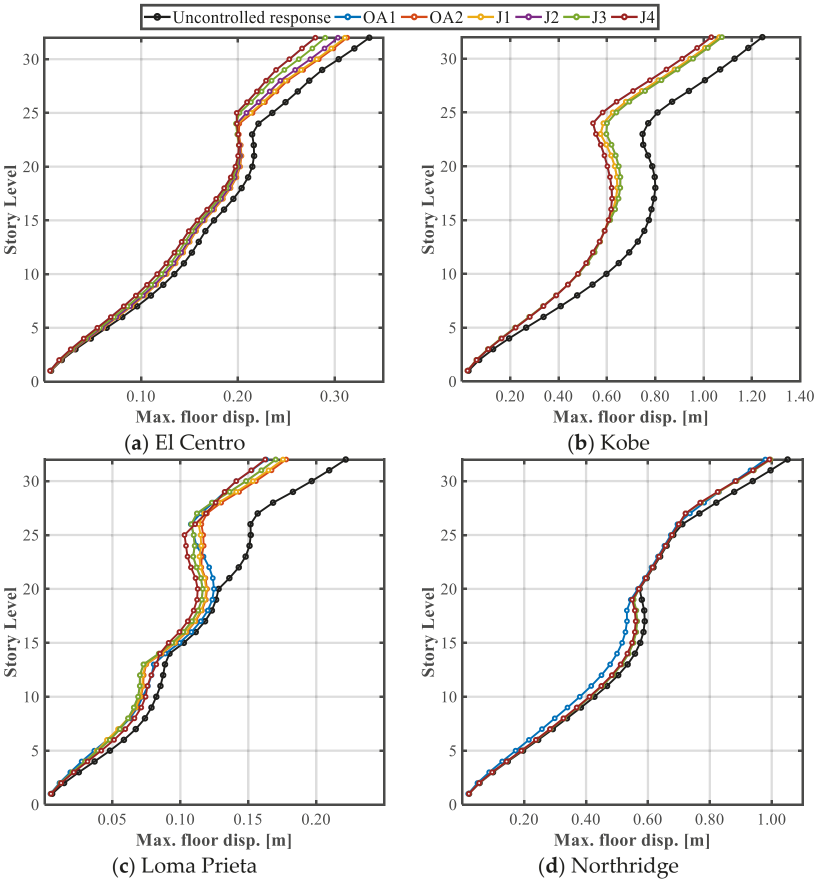


(**a**) El Centro (**b**) Kobe

(**c**) Loma Prieta (**d**) Northridge

Figure 6. Displacement of the 32nd story of the structure equipped with TMDs with μ = 0.05.

Figures 7 and 8 present the comparison of the peak displacement for μ = 0.02 and 0.05, taking into account the six proposed objective functions. It is shown that the greatest reductions in the peak displacement at each story-level are achieved using the devices with the highest mass ratio, as reported in the literature [30–50]. Moreover, the greatest reductions are observed at the higher story-levels of the structure. This behavior can be better understood by analyzing the proposed objective functions, which are mainly focused on diminishing the dynamic response at the 32nd story of the building. Hence, the optimization process influences indirectly the response parameters of the floors near the upper level.

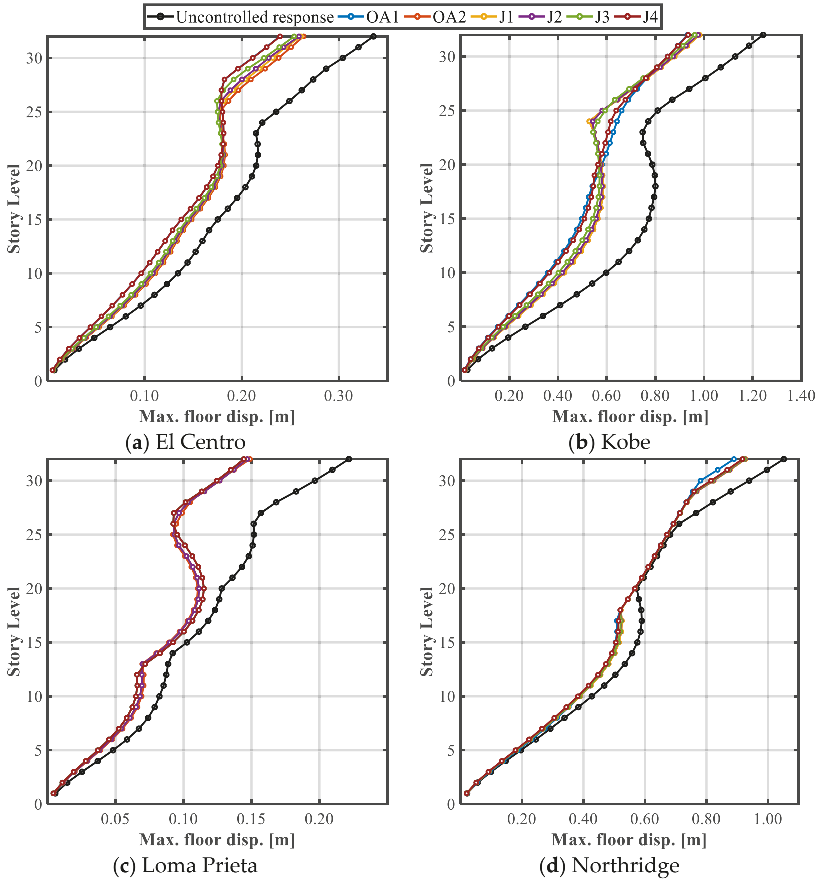


(**a**) El Centro (**b**) Kobe

(**c**) Loma Prieta (**d**) Northridge

Figure 7. Maximum floor displacement at each story level of the structure equipped with TMDs with $\mu = 0.02$.

The numerical results show that, using $\mu = 0.02$ TMD, the reduction of the maximum displacements of each floor of the structure is more effective using Kobe and Loma Prieta accelerograms than El Centro and Northridge cases. Moreover, it may be affirmed that the greatest reduction in displacement is approximately 32.13%, reached at the 25th story-level of the structure when it is controlled by the TMD designed by the J_4 approach and using the Loma Prieta earthquake excitation. This reduction shows consistency, since the J_4 approach gives the greatest possible weight to the reduction of the displacement peak over the RMS displacement values.



(**a**) El Centro (**b**) Kobe (**c**) Loma Prieta (**d**) Northridge

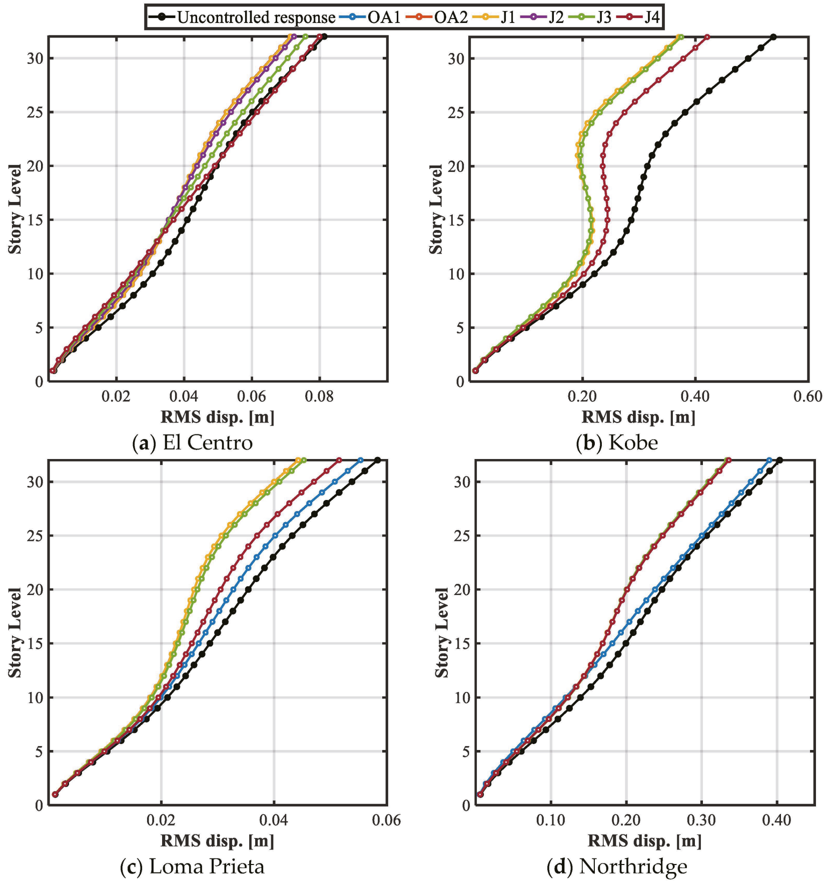
Figure 8. Maximum floor displacement at each story level of the structure equipped with TMDs with μ = 0.05.

The greatest reductions in the horizontal peak displacement at the top-story level are attained using TMDs designed through the objective functions OA1 and J_4. Using the El Centro earthquake, the horizontal peak displacement is reduced from 0.336 m to 0.280 m (16.60% reduction) using the function OA1 and J_4 indistinctly; similar results are observed using the Kobe earthquake, where the peak displacement decreases from 1.243 m to 1.032 m (17% reduction). Now, under the Loma Prieta accelerograms, the maximum displacement reduces from 0.222 m to 0.162 m (27% reduction) using the objective function J_4, while, for the Northridge excitation, the maximum displacement is decreased from 1.051 m to 0.980 m (7% reduction) with the OA1 approach.

The controlled response of the structure using optimally designed TMDs with μ = 0.05 is remarkable. The best performances are achieved with devices optimized through OA1 and J_4 approaches, which are mainly focused on reducing the peak displacements, showing reductions of up to 43 and 42%, respectively. Nevertheless, the reductions in the maximum displacements at each floor achieved by the devices designed with the remaining objective functions are notable as well, exhibiting reductions up to 39, 38, 38, and 40% for the cases of optimization OA2, J_1, J_2, and J_3, respectively. Once more, using the Northridge earthquake, the smallest reductions obtained ranged from 12 to 17% at the higher story-levels of the structure. Besides, for the three remaining accelerograms, the reductions in the maximum floor displacement oscillate between 20 and 43%.

Furthermore, the dynamic response of displacements observed in Figures 7 and 8 denotes an S-shape trend. The curve shows a linear behavior between 1st and 20th story-levels; then, there is a setback in the curve between 27th and 32nd story-levels, where it becomes linear again, although with a different inclination from the first segment representing the greatest response reductions. This behavior is closely related to the whipping lash effect that has been previously observed in high rise buildings [66]. This behavior is largely caused by changes in the compressive strength in the concrete used in the column sections of the building, which causes changes in stiffness and the dynamic behavior of the structural system.

On the other hand, Figures 9 and 10 present the behavior of the RMS values of the displacements at each level of the building controlled by TMDs with mass ratios $\mu = 0.02$ and $\mu = 0.05$, respectively.



(**a**) El Centro (**b**) Kobe

(**c**) Loma Prieta (**d**) Northridge

Figure 9. RMS displacement at each story level of the structure equipped with TMDs with $\mu = 0.02$.

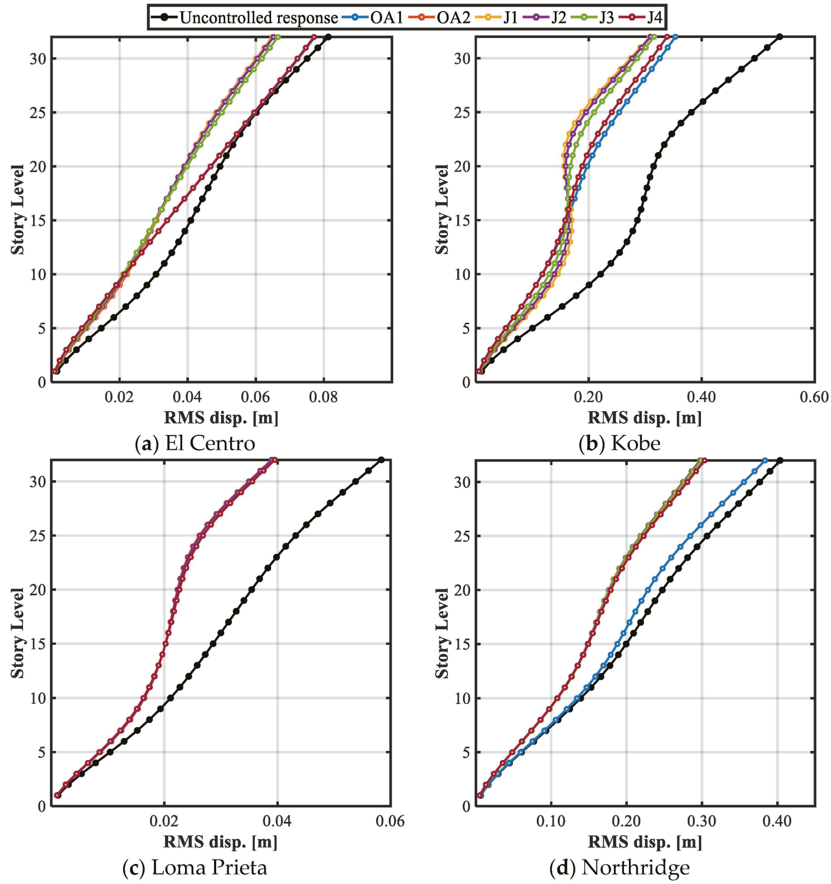
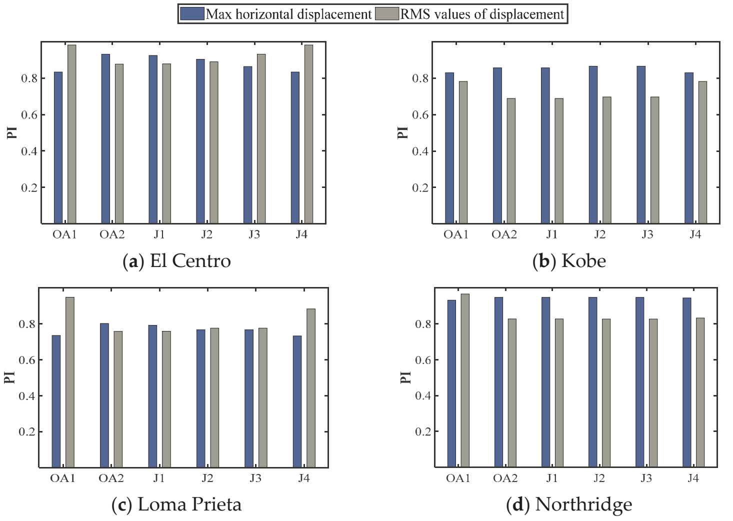


Figure 10. RMS displacement at each story level of the structure equipped with TMDs with $\mu = 0.05$.

Figure 9 shows reductions fluctuating from 19% (Northridge excitation at 23rd story-level) and 42% (Kobe excitation at 23rd story-level). Besides, the reductions in the RMS response of displacement of the roof story-level are 12, 31, 24, and 17% for the El Centro, Kobe, Loma Prieta, and Northridge accelerograms, respectively. The analysis of the RMS response of displacement curves at each floor allows the establishment that the control devices designed through the OA2, J_1, and J_2 approaches present the best performances when the RMS of displacements is analyzed; accordingly, TMDs designed with objective functions OA1 and J_4 achieve lower reductions.

The best reductions in the RMS values of the displacements at each level of the structure are obtained when the building is equipped with TMDs with $\mu = 0.05$. These reductions are attained using the OA2 and J_1 approaches: 20, 42, 33, and 26% for the El Centro, Kobe, Loma Prieta, and Northridge accelerograms, respectively. Among these reductions, the most outstanding is attained under Kobe excitation, since the reductions between 15th to 32nd story-levels ranged between 40% (0.286 m to 0.170 m at 15th story-level) and 52% (0.334 m to 0.159 m at the 22nd story-level).

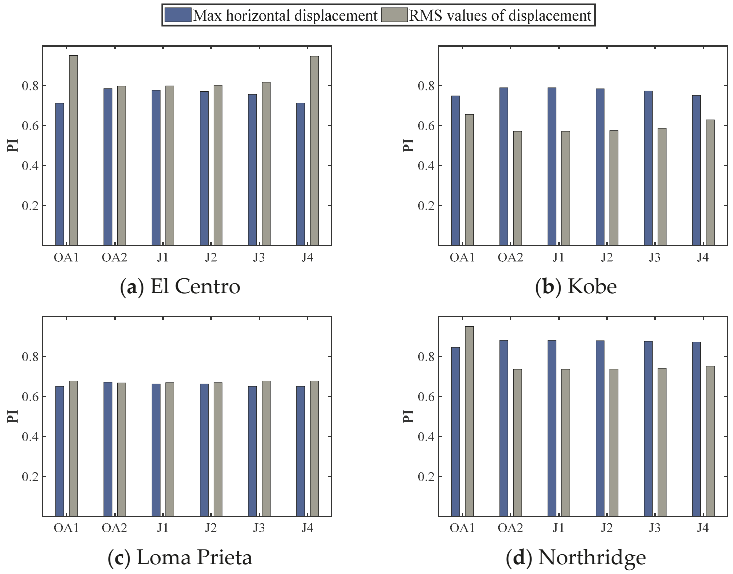
It can be noticed that the reductions in the RMS response of displacement at each floor of the structure exhibit a much more uniform and efficient behavior than the reductions of the maximum horizontal peak floor displacements, especially with $\mu = 0.05$ TMDs, and objective functions OA2, which focus on reducing the RMS displacements, and J_1, which gives the greatest weight to this parameter. Using the Kobe and Northridge records, the results report the highest response magnitudes and, at the same time, the TMDs exhibit the greatest performance. This behavior suggests that the objective functions for the optimal design of TMDs should be focused on minimizing the RMS values of displacements, or, instead, a linear combination in which the greatest weight is given to this parameter should be used. To verify this statement, Figures 11 and 12 present a comparative analysis of the six objective functions and the attained PI.



(**a**) El Centro (**b**) Kobe

(**c**) Loma Prieta (**d**) Northridge

Figure 11. PI values for every optimization approach using TMDs with μ = 0.02.



(**a**) El Centro (**b**) Kobe

(**c**) Loma Prieta (**d**) Northridge

Figure 12. PI values for every optimization approach using TMDs with μ = 0.05.

Figures 11 and 12 demonstrate that objective functions OA2, J_1, and J_2 lead to the most efficient design parameters for TMDs. Consequently, TMDs designed through these objective functions are the ones that best improve the seismic performance of the case-study by given major importance to reducing the RMS response of displacements. Furthermore, the PI values of the TMDs designed through OA2, J_1, and J_2 demonstrates that the control of the RMS values of displacements in the structure is the best possible within both evaluated parameters, which implies significant reductions in the dynamic response over time. Moreover, the horizontal peak response of displacements is markedly reduced as well. Hence, according to these results, it may be claimed that the objective function J_1 is the most balanced and effective among the six objective functions analyzed in this paper, despite the small differences in performance with OA2 and J_2.

4.5. Practical Design Recommendations

Once the optimal design parameters have been determined for each benchmark record, an attempt is made to establish a single set of optimal design parameters to work properly under any acceleration record. In that sense, the methodology proposed by Fallah and Zamiri in [60] is implemented for that purpose. This methodology establishes average design parameters according to the weighted response reduction percentages obtained previously. Therefore, the design parameters obtained from the objective function J_1, which was the most effective to control the dynamic response of displacement, are replaced in Equations (23) and (24):

$$\zeta_{d\ \mathrm{avg}} = \frac{\sum_{i=1}^{4}\left(\zeta_{opt} * R_{\mathrm{mfd}} * R_{\mathrm{RMS}}\right)}{\sum_{i=1}^{4}\left(R_{\mathrm{mfd}} * R_{\mathrm{RMS}}\right)} \tag{23}$$

$$f_{avg} = \frac{\sum_{i=1}^{4}\left(f_{opt} * R_{\mathrm{mfd}} * R_{\mathrm{RMS}}\right)}{\sum_{i=1}^{4}\left(R_{\mathrm{mfd}} * R_{\mathrm{RMS}}\right)} \tag{24}$$

where R_{mfd} and R_{RMS} denote, respectively, the reduction of the maximum floor displacement and the reduction of the maximum RMS displacement defined according to Equations (25) and (26). In addition, the counter i in the sum numbers the accelerograms used in the tuning process. More records in the optimization process will enhance the accuracy of the average design variables; however, for this example, only the numerical results computed from the records in Table 1 are considered. The computed set of average design parameters $\zeta_{d\ avg}$ and $f_{\ avg}$ are reported in Table 5.

$$R_{\mathrm{mfd}} = \left(1 - \frac{\max(|x_n|)}{\max\ (|x_n^*|)}\right) \tag{25}$$

$$R_{\mathrm{RMS}} = \left(1 - \frac{\max(\mathrm{RMS}(x_n))}{\max\ (\mathrm{RMS}(x_n^*))}\right) \tag{26}$$

Table 5. Average design parameters for TMDs with μ = 0.02 and μ = 0.05.

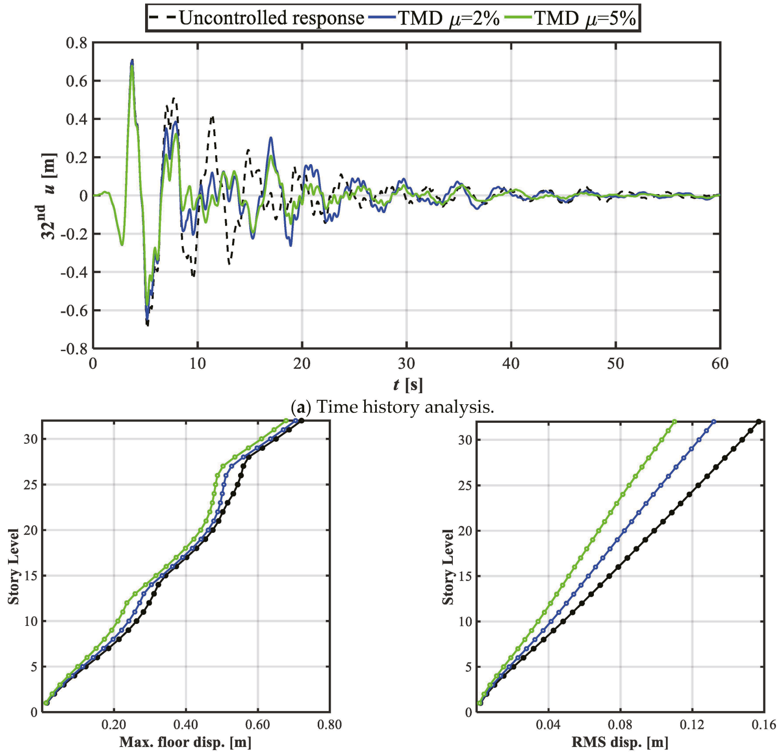
μ	$\zeta_{d\ avg}$	$f_{\ avg}$
0.02	0.103	0.967
0.05	0.209	0.929

Thereafter, the case-study is subjected to the action of the Petrolia and San Fernando acceleration records, whose details are presented in Table 6. The main purpose of using acceleration records different from the benchmark records used in the optimization process is to verify the performance of the structure and the robustness of the WOA methodology under any type of random excitation.

Table 6. Details of the new ground-motion records used.

Record	Event Name	Station	Component	PGA (g)	Duration (s)
1	Petrolia	Cape Mendocino	90 DEG	0.66	59.98
2	San Fernando	Pacoima	S74W	1.08	41.74

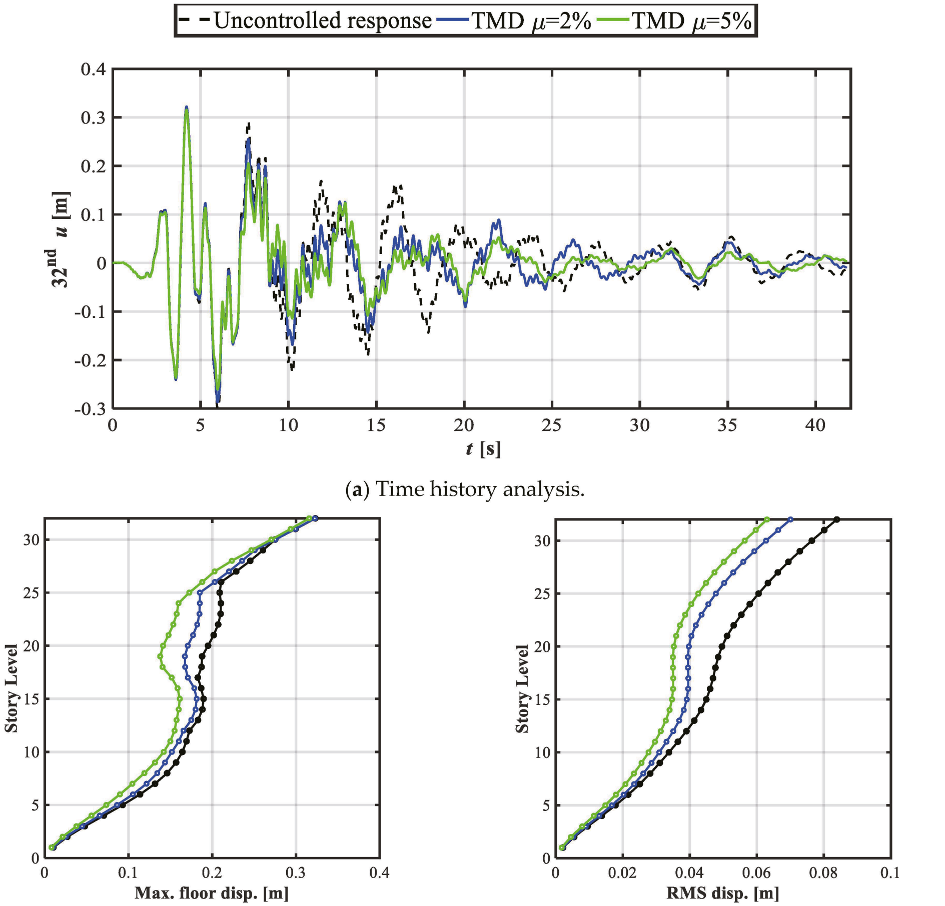
Figures 13 and 14 illustrate the response of the structure with TMDs designed with the average design parameters subjected to Petrolia and San Fernando earthquakes, respectively. Similarly, Tables 7 and 8 show the response reduction on the 32nd floor of the building. As it can be seen, the behavior of the structure controlled via the TMDs with the average design parameters exhibits notable reductions in the horizontal peak displacements and the RMS response of displacements. The decrease in the response is especially significant in the RMS values of displacement, which means that the use of the weighted procedure could be potentially applied for the design of TMDs in structures subjected to earthquake loads. Furthermore, future works should consider a large number of seismic records in order to obtain a more reliable set of design parameters that best fit a wide range of possible ground motions.



(**a**) Time history analysis.

(**b**) Maximum floor displacement at each story level (**c**) RMS values of displacement at each story level

Figure 13. Performance of the structure equipped with TMDs with average design parameters subjected to Petrolia earthquake.



(**a**) Time history analysis.

(**b**) Maximum floor displacement at each story level (**c**) RMS displacement values at each story level

Figure 14. Performance of the structure equipped with TMDs with average design parameters subjected to San Fernando earthquake.

Table 7. Response parameters for the 32nd story of the structure subjected to Petrolia earthquake.

TMD μ = 2%					
Max. Displacement			**RMS Displacement**		
Uncontrolled (m)	Controlled (m)	PI	Uncontrolled (m)	Controlled m)	PI
0.7213	0.7047	0.9770	0.1569	0.1319	0.8406
TMD μ = 5%					
Uncontrolled (m)	Controlled (m)	PI	Uncontrolled (m)	Controlled m)	PI
0.7213	0.6781	0.9401	0.1569	0.1100	0.7013

Table 8. Response parameters for the 32nd story of the structure subjected to San Fernando earthquake.

TMD μ = 2%					
Max. Displacement			**RMS Displacement**		
Uncontrolled (m)	Controlled (m)	PI	Uncontrolled (m)	Controlled (m)	PI
0.3231	0.3225	0.9980	0.0839	0.0701	0.8354
TMD μ = 5%					
Uncontrolled (m)	Controlled (m)	PI	Uncontrolled (m)	Controlled (m)	PI
0.3231	0.3158	0.9772	0.0839	0.0630	0.7513

Reductions up to 6 and 30% are observed in horizontal peak displacements and RMS displacements, respectively, using the TMD designed with a unique set of parameters derived from the methodology described in [60]. Although the methodology was originally proposed for base isolation systems, it can be adapted for the tuning of linear mass dampers since, for both base isolation systems and TMDs, optimal design variables must be determined. At first glance, the reductions attained with the average design values are not substantial. However, in earthquake engineering practice, it is impossible to know the ground motion excitation that will affect the structural systems, and, accordingly, the Petrolia and San Fernando Earthquake present dynamic properties different from the records for which the TMD was originally tuned. Thus, although small, these results validate the robustness of the optimization based on WOA and demonstrated the efficiency of the algorithm over other metaheuristics, like DEM and conventional tuning methodologies based on closed-form expressions. Finally, to achieve a realistic tuning, other critical variables should be considered, and a more comprehensive analysis is required (e.g., analysis on the stochastic nature of the seismic site conditions).

5. Conclusions

In this study, a methodology for the selection of optimal design parameters for TMDs based on WOA to reduce the dynamic response of buildings subjected to seismic excitations was presented. Six different objective functions were proposed to be minimized in the WOA methodology. These functions were related to the reduction of the maximum horizontal peak displacement of the structure, the RMS response of displacements, and four more weighted linear combinations of the same parameters. The proposed methodology was applied to a 32-story case-study derived from an actual building structure, and subjected to different ground motion records.

According to the analysis on the dynamic history of displacements of the structure, it may be concluded that the J_1 optimization approach, which gives 20% weight to the peak floor displacements and 80% weight to the RMS response of displacements, led to the best reductions in the transient response within both evaluated parameters, although a good performance was also exhibited using OA2 and J_2 optimization approaches. Furthermore, a weighted procedure for determining an average design set of parameters for TMDs was explored. The methodology, proposed originally for base isolation systems, was adapted for the tuning of linear mass dampers, showing promising results that can be improved in forthcoming investigations by considering a large number of records in the tuning process. The proposed technique for obtaining design parameters of TMDs based on WOA is demonstrated to be successful, fast, and reliable by achieving percentages of reduction in the maximum floor displacement of up to 43% and RMS values of displacement of up to 52%.

The results of this research highlighted the advantages of the WOA over other well-known metaheuristics, such as DEM and other bio-inspired algorithms. The proposed technique exhibited less computational cost, avoiding operations like mutation or crossover, and reductions in the processing time up to 45% were observed. Finally, to perform a realistic tuning of TMDs through WOA, a more comprehensive analysis is required. Future works should take into account inter-story drifts and floor accelerations which are directly associated with the structural damage, as well as consider the nonlinear properties of the structural system. Further, the stochastic analysis of the seismic signal should also be considered by developing a sensitivity analysis (e.g., a greater number of records with greater probability of occurrence and its effects on the structure must be analyzed) and reproducing synthetic accelerograms based on the damping and frequency site conditions to be used in the tuning process via WOA.

Author Contributions: Conceptualization, L.A.L.-V. and D.C.; methodology, L.A.L.-V. and D.C.; software, L.A.L.-V. and D.C.; validation, D.C.; formal analysis, L.A.L.-V. and D.C.; investigation, L.A.L.-V. and D.C.; resources, L.A.L.-V., D.C. and Y.V.-G.; writing—original draft preparation, D.C.; writing—review and editing, L.A.L.-V., D.C. and Y.V.-G.; visualization, L.A.L.-V., D.C. and Y.V.-G.; supervision, L.A.L.-V. and Y.V.-G. All authors have read and agreed to the published version of the manuscript.

Funding: The research and publication of this study was funded by Convocatoria para el Apoyo a Proyectos de Investigación y Creación Artística en la Universidad Nacional de Colombia-Sede Medellín 2020 (grant no. 49741).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data included in this study are available upon request by contact with the corresponding author.

Acknowledgments: The authors wish to thank Universidad Nacional de Colombia Sede Medellín for the financial aid and support of this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Buckle, I.G. Passive control of structures for seismic loads. *Bull. N. Z. Soc. Earthq. Eng.* **2000**, *33*, 209–221. [CrossRef]
2. Spencer, B.F.; Nagarajaiah, S. State of the Art of Structural Control. *J. Struct. Eng.* **2003**, *129*, 845–856. [CrossRef]
3. Cimellaro, G.P.; Marasco, S. Tuned-mass dampers. In *Introduction to Dynamics of Structures and Earthquake Engineering*; Springer: Cham, Germany, 2018; pp. 421–438.
4. Frahm, H. Device for Damping Vibrations of Bodies. U.S. Patent No. 989,958, 18 April 1911.
5. Ormondroyd, J.; Hartog, J.P.D. Theory of the dynamic vibration absorber. *Trans. Am. Soc. Mech. Eng.* **1928**, *50*, 9–22.
6. den Hartog, J.P. *Mechanical Vibrations*; Dover Publications, Inc.: New York, NY, USA, 1985.
7. Warburton, G.B. Optimum absorber parameters for minimizing vibration response. *Earthq. Eng. Struct. Dyn.* **1981**, *9*, 251–262. [CrossRef]
8. Warburton, G.B. Optimum absorber parameters for various combinations of response and excitation parameters. *Earthq. Eng. Struct. Dyn.* **1982**, *10*, 381–401. [CrossRef]
9. Sadek, F.; Mohraz, B.; Taylor, A.W.; Chung, R.M. A method of estimating the parameters of tuned mass dampers for seismic applications. *Earthq. Eng. Struct. Dyn.* **1997**, *26*, 617–635. [CrossRef]
10. Holmes, J. Listing of installations. *Eng. Struct.* **1995**, *17*, 676–678. [CrossRef]
11. Soto, M.G.; Adeli, H. Tuned mass dampers. *Arch. Comput. Methods Eng.* **2013**, *20*, 419–431. [CrossRef]
12. Gerges, R.R.; Vickery, B.J. Optimum design of pendulum-type tuned mass dampers. *Struct. Des. Tall Spéc. Build.* **2005**, *14*, 353–368. [CrossRef]
13. Setareh, M.; Ritchey, J.K.; Baxter, A.J.; Murray, T.M. Pendulum Tuned Mass Dampers for Floor Vibration Control. *J. Perform. Constr. Facil.* **2006**, *20*, 64–73. [CrossRef]
14. Lourenco, R. Design Construction and Testing of an Adaptive Pendulum Tuned Mass Damper. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2011.
15. Sakai, F. Tuned liquid column damper-new type device for suppression of building vibration. In Proceedings of the 1st International Conference on High-rise Buildings, Nanjing, China, 25–27 March 1989; pp. 926–931.
16. Mondal, J.; Nimmala, H.; Abdulla, S.; Tafreshi, R. Tuned liquid damper. In Proceedings of the 3rd International Conference on Mechanical Engineering and Mechatronics, Prague, Czech Republic, 14–15 August 2014. Paper. No. 68.

17. Di Matteo, A.; Furtmüller, T.; Adam, C.; Pirrotta, A. Optimal design of tuned liquid column dampers for seismic response control of base-isolated structures. *Acta Mech.* **2017**, *229*, 437–454. [CrossRef]
18. Almazán, J.L.; De la Llera, J.C.; Inaudi, J.A.; López-García, D.; Izquierdo, L.E. A bidirectional and homogeneous tuned mass damper: A new device for passive control of vibrations. *Eng. Struct.* **2007**, *29*, 1548–1560. [CrossRef]
19. Aldemir, U. Optimal control of structures with semiactive-tuned mass dampers. *J. Sound Vib.* **2003**, *266*, 847–874. [CrossRef]
20. Pourzeynali, S.; Lavasani, H.; Modarayi, A. Active control of high rise building structures using fuzzy logic and genetic algorithms. *Eng. Struct.* **2007**, *29*, 346–357. [CrossRef]
21. Lavasani, S.H.H.; Doroudi, R. Meta heuristic active and semi-active control systems of high-rise building. *Int. J. Struct. Eng.* **2020**, *10*, 232–253. [CrossRef]
22. Kaynia, A.M.; Veneziano, D.; Biggs, J.M. Seismic Effectiveness of Tuned Mass Dampers. *J. Struct. Div.* **1981**, *107*, 1465–1484. [CrossRef]
23. Sladek, J.R.; Klinger, R.E. Effect of tune mass dampers of seismic response. *J. Struct. Eng. ASCE* **1983**, *109*, 2004–2009. [CrossRef]
24. Alexander, N.A.; Schilder, F. Exploring the performance of a nonlinear tuned mass damper. *J. Sound Vib.* **2009**, *319*, 445–462. [CrossRef]
25. Li, L.; Du, Y. Design of Nonlinear Tuned Mass Damper by Using the Harmonic Balance Method. *J. Eng. Mech.* **2020**, *146*, 04020056. [CrossRef]
26. Nigdeli, S.M.; Bekdaş, G. Optimum design of multiple positioned tuned mass dampers for structures constrained with axial force capacity. *Struct. Des. Tall Spéc. Build.* **2019**, *28*, e1593. [CrossRef]
27. Marian, L.; Giaralis, A. Optimal design of a novel tuned mass-damper–inerter (TMDI) passive vibration control configuration for stochastically support-excited structural systems. *Probabilistic Eng. Mech.* **2014**, *38*, 156–164. [CrossRef]
28. Giaralis, A.; Taflanidis, A. Optimal tuned mass-damper-inerter (TMDI) design for seismically excited MDOF structures with model uncertainties based on reliability criteria. *Struct. Control. Health Monit.* **2018**, *25*, e2082. [CrossRef]
29. Ruiz, R.; Taflanidis, A.; Giaralis, A.; Lopez-Garcia, D. Risk-informed optimization of the tuned mass-damper-inerter (TMDI) for the seismic protection of multi-storey building structures. *Eng. Struct.* **2018**, *177*, 836–850. [CrossRef]
30. Salvi, J.; Rizzi, E. Optimum tuning of Tuned Mass Dampers for frame structures under earthquake excitation. *Struct. Control. Heal. Monit.* **2015**, *22*, 707–725. [CrossRef]
31. Islam, M.S.; Do, J.; Kim, D. Multi-Objective Optimization of TMD for Frame Structure Based on Response Surface Methodology and Weighted Desirability Function. *KSCE J. Civ. Eng.* **2017**, *22*, 3015–3027. [CrossRef]
32. Salvi, J.; Rizzi, E.; Rustighi, E.; Ferguson, N.S. Optimum Tuning of Passive Tuned Mass Dampers for the Mitigation of Pulse-Like Responses. *J. Vib. Acoust.* **2018**, *140*, 061014. [CrossRef]
33. Leung, A.Y.T.; Zhang, H.; Cheng, C.C.; Lee, Y.Y.R. Particle swarm optimization of TMD by non-stationary base excitation during earthquake. *Earthq. Eng. Struct. Dyn.* **2008**, *37*, 1223–1246. [CrossRef]
34. Leung, A.; Zhang, H. Particle swarm optimization of tuned mass dampers. *Eng. Struct.* **2009**, *31*, 715–728. [CrossRef]
35. Bekdaş, G.; Nigdeli, S.M. Estimating optimum parameters of tuned mass dampers using harmony search. *Eng. Struct.* **2011**, *33*, 2716–2723. [CrossRef]
36. Bekdaş, G.; Nigdeli, S.M. Mass ratio factor for optimum tuned mass damper strategies. *Int. J. Mech. Sci.* **2013**, *71*, 68–84. [CrossRef]
37. Nigdeli, S.M.; Bekdaş, G. Optimum tuned mass damper design in frequency domain for structures. *KSCE J. Civ. Eng.* **2017**, *21*, 912–922. [CrossRef]
38. Bekdaş, G.; Nigdeli, S.M. Optimization of tuned mass damper with harmony search. In *Metaheuristic Applications in Structures and Infrastructures*; Gandomi, A.H., Yang, X.-S., Alavi, A.H., Talatahari, S., Eds.; Elsevier: Amsterdam, The Netherlands, 2013; pp. 345–371.
39. Nigdeli, S.M.; Bekdaş, G. Tuning and position optimization of mass dampers for seismic structures. In Proceedings of the International Conference on Harmony Search Algorithm Springer, Singapore, 22–24 February 2017; pp. 232–239.
40. Bekdaş, G.; Nigdeli, S.M.; Yang, X.S. Metaheuristic based optimization for tuned mass dampers using frequency domain responses. In Proceedings of the International Conference on Harmony Search Algorithm Springer, Singapore, 22–24 February 2017; pp. 271–279.
41. Farshidianfar, A.; Soheili, S. Optimization of TMD parameters for earthquake vibrations of tall buildings including soil structure interaction. *Iran Univ. Sci. Technol.* **2013**, *3*, 409–429.
42. Farshidianfar, A.; Soheili, S. Ant colony optimization of tuned mass dampers for earthquake oscillations of high-rise structures including soil-structure interaction. *Soil Dyn. Earthq. Struct.* **2013**, *51*, 14–22. [CrossRef]
43. Nigdeli, S.M.; Bekdaş, G.; Yang, X.S. Optimum tuning of mass dampers for seismic structures using flower pollination algorithm. *Int. J. Theor. Appl. Mech.* **2016**, *1*, 264–268.
44. Nigdeli, S.M.; Bekdaş, G.; Yang, X.S. Optimum tuning of mass dampers by using a hybrid method using harmony search and flower pollination algorithm. In Proceedings of the International Conference on Harmony Search Algorithm Springer, Singapore, 22–24 February 2017; pp. 222–231.
45. Yucel, M.; Bekdaş, G.; Nigdeli, S.M.; Sevgen, S. Estimation of optimum tuned mass damper parameters via machine learning. *J. Build. Eng.* **2019**, *26*, 100847. [CrossRef]
46. Bekdaş, G.; Nigdeli, S.M.; Yang, X.-S. A novel bat algorithm based optimum tuning of mass dampers for improving the seismic safety of structures. *Eng. Struct.* **2018**, *159*, 89–98. [CrossRef]

47. Etedali, S.; Rakhshani, H. Optimum design of tuned mass dampers using multi-objective cuckoo search for buildings under seismic excitations. *Alex. Eng. J.* **2018**, *57*, 3205–3218. [CrossRef]
48. Pourzeynali, S.; Salimi, S.; Kalesar, H.E. Robust multi-objective optimization design of TMD control device to reduce tall building responses against earthquake excitations using genetic algorithms. *Sci. Iran.* **2013**, *20*, 207–221.
49. Pourzeynali, S.; Salimi, S. Multi-objective optimization design of control devices to suppress tall buildings vibrations against earthquake excitations using fuzzy logic and genetic algorithms. In *Design Optimization of Active and Passive Structural Control Systems*; Lagaros, N.D., Ed.; IGI Global: Hershey, PA, USA, 2015; pp. 180–215.
50. Pal, S.; Singh, D.; Kumar, V. Hybrid SOMA: A tool for optimizing TMD parameters. In Proceedings of the Sixth International Conference on Soft Computing for Problem Solving, Singapore, 23–24 December 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 35–41.
51. Kaveh, A.; Javadi, S.; Moghanni, R.M. Optimal structural control of tall buildings using tuned mass dampers via chaotic optimization algorithm. *Structures* **2020**, *28*, 2704–2713. [CrossRef]
52. Bekdaş, G.; Nigdeli, S.M. Metaheuristic based optimization of tuned mass dampers under earthquake excitation by considering soil-structure interaction. *Soil Dyn. Earthq. Eng.* **2017**, *92*, 443–461. [CrossRef]
53. Salvi, J.; Pioldi, F.; Rizzi, E. Optimum Tuned Mass Dampers under seismic Soil-Structure Interaction. *Soil Dyn. Earthq. Eng.* **2018**, *114*, 576–597. [CrossRef]
54. Bekdaş, G.; Nigdeli, S.M. An Optimum Tuning Application of Mass Dampers Considering Soil-Structure Interaction: Metaheuristic-Based Optimization of TMDs. In *Handbook of Research on Predictive Modeling and Optimization Methods in Science and Engineering*; Kim, D., Sekhar Roy, S., Länsivaara, T., Deo, R., Samui, P., Eds.; IGI Global: Hershey, PA, USA, 2018; pp. 44–73.
55. Bekdaş, G.; Kayabekir, A.E.; Nigdeli, S.M.; Toklu, Y.C. Transfer function amplitude minimization for structures with tuned mass dampers considering soil-structure interaction. *Soil Dyn. Earthq. Eng.* **2019**, *116*, 552–562. [CrossRef]
56. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
57. Storn, R.; Price, K.V. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
58. Caicedo, D.; Lara-Valencia, L.; Blandon, J.; Farbiarz, Y. Differential Evolution for the Optimal Design of Tuned Mass Dampers in a Building of Medellin City. In Proceedings of the 17th World Conference on Earthquake Engineering, Sendai, Japan, 27 September–2 October 2020.
59. Caicedo, D.; Lara-Valencia, L.; Blandon, J.; Graciano, C. Seismic response of high-rise buildings through metaheuristic-based optimization using tuned mass dampers and tuned mass dampers inerter. *J. Build. Eng.* **2021**, *34*, 101927. [CrossRef]
60. Fallah, N.; Zamiri, G. Multi-objective optimal design of sliding base isolation using genetic algorithm. *Sci. Iran.* **2012**, *20*, 87–96. [CrossRef]
61. Kaveh, A.; Ghazaan, M.I. Enhanced whale optimization algorithm for sizing optimization of skeletal structures. *Mech. Base. Des. Struct. Mach.* **2017**, *45*, 345–362. [CrossRef]
62. Chen, H.; Xu, Y.; Wang, M.; Zhao, X. A balanced whale optimization algorithm for constrained engineering design problems. *Appl. Math. Model.* **2019**, *71*, 45–59. [CrossRef]
63. Azizi, M.; Ejlali, R.G.; Ghasemi, S.A.M.; Talatahari, S. Upgraded whale optimization algorithm for fuzzy logic based vibration control of nonlinear steel structure. *Eng. Struct.* **2019**, *192*, 53–70. [CrossRef]
64. Pacific Earthquake Engineering Research Center: Ground Motion Database. Available online: http://peer.berkeley.edu/peer_ground_motion_database (accessed on 11 November 2019).
65. Lara-Valencia, L.; Caicedo, D.; Farbiarz, Y.; Brito, J.; Valencia-Gonzalez, Y. Design of a Tuned Mass Damper Inerter using optimization based on exhaustive search for the vibration control of a seismically excited structure. In Proceedings of the 17th World Conference on Earthquake Engineering, Sendai, Japan, 27 September–2 October 2020.
66. Yang, L.J.; Liu, D.W.; Guo, Z.L.; Li, J.; Dai, B.H. Engineering Mechanics in Whipping Effect of High-Rise Building. *Appl. Mech. Mater.* **2014**, *540*, 173–176. [CrossRef]

symmetry

MDPI

Review

Challenges and Open Problems of Legal Document Anonymization

Gergely Márk Csányi [1,2], Dániel Nagy [1], Renátó Vági [1,3], János Pál Vadász [1,4] and Tamás Orosz [1,*]

[1] MONTANA Knowledge Management Ltd., H-1097 Budapest, Hungary; csanyi.gergely@montana.hu (G.M.C.); nagy.daniel@montana.hu (D.N.); vagi.renato@montana.hu (R.V.); vadasz.pal@montana.hu (J.P.V.)
[2] Department of Electric Power Engineering, Budapest University of Technology and Economics, H-1111 Budapest, Hungary
[3] Doctoral School of Law, Eötvös Loránd University, H-1053 Budapest, Hungary
[4] Institute of the Information Society, National University of Public Service, H-1083 Budapest, Hungary
* Correspondence: orosz.tamas@montana.hu

Abstract: Data sharing is a central aspect of judicial systems. The openly accessible documents can make the judiciary system more transparent. On the other hand, the published legal documents can contain much sensitive information about the involved persons or companies. For this reason, the anonymization of these documents is obligatory to prevent privacy breaches. General Data Protection Regulation (GDPR) and other modern privacy-protecting regulations have strict definitions of private data containing direct and indirect identifiers. In legal documents, there is a wide range of attributes regarding the involved parties. Moreover, legal documents can contain additional information about the relations between the involved parties and rare events. Hence, the personal data can be represented by a sparse matrix of these attributes. The application of Named Entity Recognition methods is essential for a fair anonymization process but is not enough. Machine learning-based methods should be used together with anonymization models, such as differential privacy, to reduce re-identification risk. On the other hand, the information content (utility) of the text should be preserved. This paper aims to summarize and highlight the open and symmetrical problems from the fields of structured and unstructured text anonymization. The possible methods for anonymizing legal documents discussed and illustrated by case studies from the Hungarian legal practice.

Keywords: data mining; text mining; text recognition; machine learning; knowledge engineering

Citation: Csányi, G.M.; Nagy, D.; Vági, R.; Vadász, J.P.; Orosz, T. Challenges and Open Problems of Legal Document Anonymization. *Symmetry* **2021**, *13*, 1490. https://doi.org/10.3390/sym13081490

Academic Editor: Leyi Wei

Received: 16 June 2021
Accepted: 2 August 2021
Published: 13 August 2021

1. Introduction

Digitalization of judicial systems is an important goal of the European Union [1]. Sharing and making court decisions and different legal documents accessible online is a crucial part of this intention. These public databases can make the administration of justice more transparent. These openly accessible documents can also help decision-making processes and research for the legal practice [2–6]. Many novel databases provide easy access to court decisions and legal documents. These databases do not only share the digitized version of the documents, but also use different machine learning methodologies to find the connecting documents and the most relevant keywords of the documents. Some novel databases go further and connect these legal documents with other databases and provide them as a linked data service [4]. The databases built on published court decisions are very attractive for a wide range of professionals in the legal information market in order to reduce the research time.

However, this openly published data can contain much sensitive personal information which should not be published [7]. The current anonymization practice of these legal documents means the masking or removing the names or other direct identifiers from these documents [8]. Many current research projects have been published in this topic, which shows how state-of-the-art Named Entity Recognition (NER) methods can be used to achieve an automatized process, which can reduce the necessary human work, which can take up to 40 min for a single document [2,9,10].

However, these solutions result the simple pseudonymization of these documents only. General Data Protection Regulation (GDPR) makes a difference between anonymization and pseudonymization. The main difference between these methods is that during the anonymization process the anonymized data is modified irreversibly (see Figure 1) [11–13], while pseudonymization processes cannot prevent privacy breaches, as shown in the literature of medical data anonymization [14–16].

The most sophisticated tools in the legal document anonymization domain (e.g., ANOPPI and Finlex [2,5,10,17]) use state-of-the-art machine learning-based Named Entity Recognition (NER) technology to find and replace direct identifiers in judicial texts. However, without the deeper statistical analysis of the possible pseudo-identifiers, these applications cannot make an irreversible anonymization on the documents [18].

In 2019, a group of researchers carried out a linking attack against anonymized legal cases in Switzerland. They published a study where they presented that using artificial intelligence methods with big data collected from other publicly available databases, they could re-identify 84% of the people, being anonymized in this database, in less than an hour [19]. This can happen because a legal document contains a great deal of microdata from the involved participants, which can be used as quasi or pseudo identifiers to re-identify the participants by using third party databases (Figure 2) [20–22].

Furthermore, there are available methods for anonymizing medical data for research purposes [16,23] but, these anonymization methodologies have been developed for structured data, where every record contains the same kind of attributes and removing the items does not have an effect of the text understandability. For this reason, this topic is asymmetric regarding the significant difference in the behaviour of structured and unstructured data.

In case of a legal document, the processed data is unstructured text; therefore, the extracted data can be represented in a sparse matrix of different attributes [24]. It is also important that the anonymized text should remain easy to read after the pseudonymization process [25]. Let us examine the case of the following example sentence: "John and Julie went on holiday to Barbados in August 2016". Simple masking replaces every name, date and place in the text with "XXX". This results in losing essential text properties which are needed for the utilization of the text [26]. Furthermore, if a person is referred as "Iron Lady", the NER-based anonymization tools will not recognize it as a direct identifier; however, the reader will probably know from the context that Margaret Thatcher is the referred person.

Mozes and Kleinberg proposed a TILD methodology (TILD is an acronym from Technical tool evaluation, Information loss and De-anonymization) [27] for anonymizing texts. They proposed to analyze three objectives during the anonymization process together. These three critical aspects are how well the system detects the pseudo and direct identifiers, how much information is lost by removing the different parts of the sentence and to assess the possible data breaches.

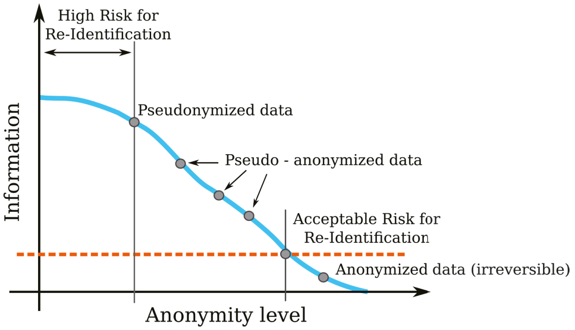


Figure 1. The difference between the anonymous and the pseudo-anonymized data. The red line represents the threshold, where the re-identification of the anonymized or anonymous data is impossible.

Theorem 1 (Pseudonymization, GDPR [28])**.** *"The processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person".*

Theorem 2 (Personal Data, GDPR [28])**.** *"Personal data is any data that could possibly related to a person directly or indirectly (Opinion 4/2007 on the concept of personal data)".*

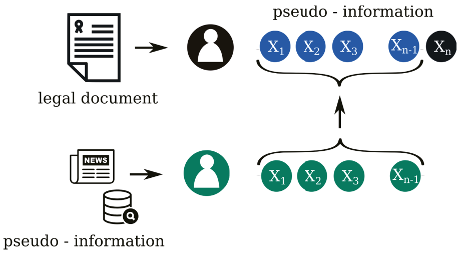


Figure 2. The image shows the found pseudo or quasi-identifiers in the text ($x_1 ... x_n$) the role of anonymization and privacy-preserving data publishing, and a possible linking attack, which uses publicly available data from newspapers and other public databases.

The main goal of the paper is to provide a thorough insight on the flaws of the current practice of judicial text anonymization processes. The current anonymization practice in many European Union countries means the masking of the names and other direct identifiers of the involved persons. This process does not fulfill the requirements of the General Data Protection Regulation. The first part of the paper summarizes the existing data anonymization techniques, which mostly support anonymizing structured data. The goal of the second part of the paper is to illustrate and highlight the importance of this topic. These examples come from the database of the decisions of the Hungarian Court [29]. These examples show that mathematical statistical analysis is important in filtering those unique events, that may serve as a primary identifier (e.g., the surgeon amputates the wrong leg). Moreover, those applications and services, which link the legal documents together with other databases, need a special care to consider the GDPR recommendations.

2. Privacy and Anonymization

The apparent goal of the data anonymization or de-identification process is to remove all of the involved individuals' direct identifiers, such as names, addresses, phone numbers, etc. [16]. This step might seem obvious; however, it is easy to defeat this kind of anonymization by linking attacks or more sophisticated attacking models [30]. These algorithms collect the text's auxiliary information and link it with public, not anonymous records [31–33], as shown in Figure 2.

Theorem 3 (Pseudo or quasi-identifiers [34,35])**.** *Attributes that do not identify a person directly but by linking to other information sources (e.g., publicly available databases, news sources) that could be used to re-identify people are called pseudo or quasi-identifiers. e.g., date of specific events (death, birth, discharge from a hospital etc.), postal codes, sex, ethnicity etc. [36–39].*

Sweeney made a famous linking attack on the set of public health records collected by The National Association of Health Data Organizations (NAHDO) in many states where they had legislative mandates to collect hospital-level data (Figure 3). These data collections were anonymized but contained several pseudo-identifiers (ZIP code, gender, ethnicity, etc.). She bought two voter registration lists from Cambridge, Massachusetts. These voter registration lists contained enough pseudo-identifiers to be linked with medical data records [31,40]. Three pieces of pseudo-information (ZIP code, gender, and date of birth) were enough to link the two databases, and Sweeney was able to re-identify about 88% of the people. She also found that about 87% of the population in the United States could be identified by using simple demographic attributes (5-digit ZIP, gender, date of birth) and about 53% when only place, gender, and date of birth is known [22,40].

Another good example would be the case study about the Netflix Prize dataset. Netflix, at that time renting DVDs online, published a dataset and offered USD 1 million as a prize to improve their movie recommendation system [20,21,41]. The dataset contained ratings for different movies and the exact dates when these ratings were made. The studies of Narayanan and Shmatikov have shown that users can be identified with about 95% probability only by knowing one's ratings for eight movies (out of which six is known correctly) and knowing the date of publishing within a 2-week-long interval [20,21]. Although, at first sight, knowing someone's taste might not seem to be dangerous at all, and in the vast majority of the cases this would be true, political orientation or religious views might be just two examples for possibly sensitive information that could be learned from one's rating history and surely not everyone would agree to share these pieces of information with the whole world [21].

A similar privacy breach on unstructured data, namely on query log data, has been reported in [42].

Figure 2 shows a basic attack model against an anonymized legal case. Suppose the aim of the curator is to preserve privacy for n pseudo-information of the legal document. However, an attacker has information about $n-1$ pseudo information of an individual except the nth record, denoted by X_n. This information of the attacker (the $n-1$ records) is also known as the background information. By making a query on the dataset of legal documents, the attacker can learn the aggregate information about n records. Hence, by comparing the query result and the background information, the attacker can easily identify the additional information of the record n. Having spotted the additional information, the attacker could move on to other datasets, performing the same operation until the victim can be re-identified. The aim of differential privacy is to protect against these kind of attacks [30,43].

Theorem 4 (Equivalence class [44])**.** *"All data records that share the same quasi-identifiers form an equivalence class". For example, all companies founded on 2nd July 2018 form an equivalence class.*

Theorem 5 (Differential privacy [30])**.** *"A mechanism M satisfies ϵ-differential privacy if, for any datasets x and y differing only on the data of a single individual and any potential outcome $\hat{q}$:*

$$\mathbb{P}[\mathcal{M}(x) = \hat{q}] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{M}(y) = \hat{q}], \tag{1}$$

where the ϵ parameter represents the the bound of the privacy loss and its value can be selected between 0 and 1. Where 0 represents that there is no privacy loss, the result of the two examined methodology is the same, no extra information is revealed from the selected individual".

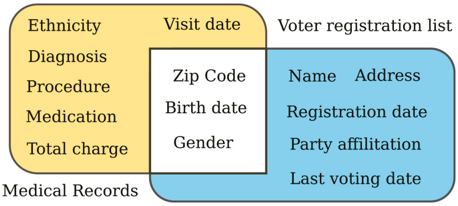


Figure 3. The image shows Sweeney's linking attack. Three pseudo identifiers were enough to re-identify the private data of many individuals.

Differential privacy is a strict, robust, and formal definition of data protection, which ensures the following three criteria: post-processing, composition, and group-privacy of the data. Post-processing means that the analysis of the published information or linking this microdata with other databases results in a set of data that satisfies the criteria of differential privacy. Therefore, it protects from the successful reconstruction or tracing from this data. Composition and group-privacy also protects privacy, i.e., when the information is shared in multiple places or with multiple individuals [30,45].

2.1. Privacy Models

Ensuring privacy while maintaining the utility of data are two directions contradicting each other. To satisfy both needs, several privacy models have been introduced. These models use generalization or suppression (i.e., deleting) of attributes in records to reach the defined privacy [16,46].

These models distinguish three types of data: direct identifiers (e.g., full name, social security number, etc.), quasi-identifiers (e.g., age, job, date of birth, etc.), and confidential attributes (e.g., religion, specific disease, salary, etc.) [47]. Privacy models assume that the directly identifying attributes have been removed and focus on quasi-identifiers and confidential attributes. There are many privacy methods represented in the literature [16,48–50]. In this section, we present some widely used methods: k-anonymity [51], l-diversity [52], and t-closeness [53].

2.1.1. k-Anonymity

A widely known approach is k-anonymity, which has been introduced by Samarati and Sweeney [51,54–56]. The formal definition can be seen in Definition 6.

Theorem 6 (k-anonymity [51,54–56])**.** *"A dataset satisfies k-anonymity for $k > 1$ when at least k records exist in the dataset for each combination of quasi-identifiers".*

This privacy model does not ensure privacy in itself. K-anonymity is able to prevent identity disclosure, which makes impossible the exact mapping of the k-anonymized record to the original dataset. However, it is prone against attribute disclosure, e.g., if k pieces of k-anonymized records share the same confidential attributes [47]. As an example, if there are k number of records identified by the following attributes: Age = 55, Height = 185 cm, Sex = Male, but all records share the same sensitive information, e.g., all of them have AIDS.

2.1.2. l-Diversity

An approach to solve attribute disclosure is l-diversity [52]. The formal definition can be seen in Definition 7.

Theorem 7 (l-diversity [47,53])**.** *"An equivalence class has l-diversity if there are at least l "well-represented" values for the sensitive attribute. A table has l-diversity if every equivalence class of the table has l-diversity".*

The term "well-represented" is not an exact definition. Machanavajjhala et al. have given the following interpretations: distinct l-diversity, entropy l-diversity, recursive (c,l)-diversity [52].

However, Li et al. [53] pointed out that l-diversity cannot prevent disclosure in case of skewness and similarity attacks, and that achieving l-diversity may be difficult and unnecessary [47].

Similarity attack [47] can be performed when the group of sensitive attributes fulfill the criterion of l-diversity but are semantically similar, e.g., in a 3-diverse medical dataset where disease is a sensitive attribute, the values are (lung cancer, stomach cancer, skin cancer). Despite meeting the requirement of 3-diversity, it is possible to learn that someone has cancer.

Skewness attack [47] can happen when the overall distribution is skewed, in which case l-diversity cannot prevent attribute disclosure. Consider a database of 10,000 test results for a disease where being positive is sensitive information, and there are 1% positive tests. An equivalence class having 24 positive records and only one negative record would meet the criteria of distinct 2-diversity and would have higher Entropy l-diversity than the whole dataset, although anyone in the equivalence class is 96% positive rather than 1%. Oddly, the same l-diversity could be calculated when the equivalence class contains only one positive and 24 negative members, although the risks are highly different.

2.1.3. t-Closeness

Li et al. in [53] defined t-closeness (Definition 8) that provides a solution for the two vulnerabilities of l-diversity mentioned above, namely the skewness and similarity attacks.

Theorem 8 (t-closeness [53])**.** *"An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness".*

Li et al., in their work, used the Earth Mover's Distance [57] as distance metric to calculate t-closeness [53]. However, the authors claim that using other distance metrics (e.g., cosine-distance, Euclidean-distance etc.) is also possible.

Domingo et al., in [47], criticizes that although in [53] several ways have been represented to check t-closeness, no computational procedure has been given to enforce this property. Moreover, if such a procedure was available, it would cause huge harm to data utility since t-closeness destroys the correlations between quasi-identifiers and confidential attributes. The only way to prevent damage to data utility is to increase threshold t, hence relaxing t-closeness.

2.2. Available Tools, Solutions

The introduction of GDPR has increased the number of pseudonymization software or solution available. In this section, we provide a short overview regarding solutions that are connected or could be connected somehow to the anonymization in legal domain.

Vico and Calegari [58] presented a general solution for anonymizing a document in any domain and tested its functionality on legal documents, although no quantitative validation was presented in their paper. The whole solution is more akin to a generic flowchart that could be applied to any type of document. The backbone of the method is a Named Entity Recognition model. The new idea that the paper brought in is that from the results of the extraction, the entities belonging to the same person or location were assigned to each other by means of clustering, e.g., if the full name John Doe was mentioned once in the text, and afterward it is also referred to somewhere as J. Doe or John D., the entities were considered to be the same. The found entities have been then modified to generic terms. The drawback of this solution is that it does not differentiate between direct and quasi-identifiers, it only focuses on extracting direct identifiers, thus risk analysis of the remaining quasi-identifiers is also missing.

Povlsen et al. adopted a Danish NER solution to the legal domain, based on hand-crafted grammar rules, gazetteers, and lists of domain specific named entities [9]. The solution was tested on 16 pages of legal content containing 30 entities, that is a small dataset for testing. Moreover, the authors focused on identifying direct identifiers not taking quasi-identifiers into consideration during their anonymization process.

NETANOS, an open-source anonymization tool, focuses on context-preserving anonymization to maintain readability of texts. This is practically carried out by replacing the entities by their types, e.g., "John went to London with Mary". is replaced by "[PER_1] went to [LOC_1] with [PER_2]". The study offers a comparison between manual anonymization, NETANOS anonymization and UK Data Service (https://bitbucket.org/ukda/ukds.tools.textanonhelper/wiki/Home) (accessed on 1 May 2021) anonymization techniques involving hundreds of people. Authors claim that their software achieves almost the same level in possibility of re-identification as manual anonymization [25]. However, this solution also takes direct identifiers into consideration, without paying attention for quasi-identifiers.

There are many tools available for anonymization of medical records and health data, such as the UTD Anonymization Toolbox [59], μ-AND [60], the Cornell Anonymization Toolkit [61], TIAMAT [62], Anamnesia [23] or SECRETA [63]. These solutions are able to fulfill privacy criteria defined by the user automatically [16]. However, the shortcoming of these solutions is that they often support only a limited number of privacy and data transformation models [16]. ARX [64] is an open-source anonymization tool, which supports a wide range of anonymization techniques, such as k-anonymity, l-diversity, etc. These techniques were developed to provide a flexible and semi-automatic solution for anonymization of data tables [16,46,64,65]. The tool was developed for medical data that is in a database format; therefore, these software solutions cannot contain methodologies for natural language processing of unstructured texts.

HIDE is a tool developed to anonymize health data [66,67]. The tool takes into account that a significant part of health data exists in unstructured text form, e.g., clinical notes, radiology or pathology reports, and discharge summaries and extracts direct identifiers (e.g., patient name, address, etc.), and quasi-identifiers (e.g., age, zip code, etc.) and sensitive information (e.g., disease type) from the unstructured text. Since a person can have multiple health scans, the device tries to attach the information extracted from the scanned document to people existing in the database. The database thus expanded allows HIDE to perform anonymization procedures such as k-anonymity, t-closeness, l-diversity on the whole database that is a traditional relational database. Extraction is based on a Conditional random Field (CRF) NER model. The wide range of extracted data and organizing the documents to a database makes HIDE an outstanding anonymization tool. However, the solution was implemented to tackle medical documents, not legal ones, so inherently misses entities characteristic for the legal domain (e.g., events, multiple sides etc.).

ANOPPI [2], is an automatic or semi-automatic pseudonymization service for Finnish court judgments. It uses state-of-the-art, BERT-based [18] NER models alongside rule-based solutions to retrieve as much direct identifiers as possible from legal texts. However, the solution does not take into consideration the fact that other quasi-identifiers, e.g., events in themselves can be direct identifiers, or a small set of quasi-identifiers can lead to a privacy breach as we show in Section 5.4. The tool emphasizes the importance of utilizing the legal text after pseudonymization has been carried out, since Finnish is a highly inflected language (similar to Hungarian). The tool performs morphological analysis in order to apply the correct inflected form for the pseudonym, hence improving readability of pseudonymized text. The aim of the ANOPPI project is to create a general purpose anonymization tool, by removing direct identifiers. It has been shown that methods, which remove the directly identifying attributes only (i.e., names, email addresses or personal identification numbers) can not prevent privacy breaches [14–16] and would contradict the No Free Lunch Theorem in related fields [68–70].

3. Types of Privacy Attacks

Knowing the different types of privacy attacks is essential to considering and quantifying the different privacy risks. There can be different goals of the adversaries: re-identification, reconstruction or tracing of different persons, where the attacker only wants to know that the given person is connected to the given dataset or not [30]. The authors of [49,71] published three different kinds of attacker models (prosecutor, journalist and marketer), which typically considered in medical health data anonymization software solutions [16,46]. These attacking strategies assume that the adversary has different a priori information about the database or the subject of the attack. For instance, the prosecutor knows that the data about the searched person is involved in the given or the connecting cases. The journalist has some a priori information about the searched person, maybe she knows some background data, which can be linked with the microdata of the connecting cases. The marketer model assumes that the adversary has no prior knowledge, but their goal is to re-identify a large number of individuals for marketing purposes [71,72].

The previously introduced Swiss case study about re-identifying pseudonymized legal cases used a marketer strategy with a simple linking attack. The researchers wanted to re-identify as many people as they can, surprisingly, they could re-identify 84% of them within an hour) [19]. The reason for this is the presence of a large number of quasi-identifiers in legal documents. The wide range of quasi-identifiers, both in number and type, generally provide enough information for a human to de-anonymize the documents. As Mozes and Kleinberg pointed out, even a single identifying attribute can be sufficient for re-identification [27]. Since legal cases always tell the story of the two or more sides involved, the texts contain many events, people, and institutions (two sides, judges, attorneys, witnesses, etc.). A legal document contains one case, which is interpreted in at least three different ways. It is possible that each of these single interpretations cannot contain enough microdata to re-identify the persons, but the series of these interpretations serve enough information for de-anonymization (Figure 4), as it is shown in Section 5.4.

Figure 4 shows an example of how legal documents may be connected to each other or to other databases via the wide range of these quasi-identifiers. It is important to point out that the majority of the published cases are part of a “decision-chain”: decisions from the first instance up to the Supreme Court are linked together. This means that there are usually three documents being linked tightly to each other. This poses a threat of de-anonymization since it is not sufficient to have two of these documents properly anonymized if the third is not properly anonymized. Hence, the integrity of a chain must be kept. The quasi-identifiers learnt from the joined documents can be used to match these data with other publicly available databases in order to de-anonymize the parties involved.

Because many novel legal databases aim to find and publish the connecting cases, these databases can increase the risk of attacks, as mentioned above [4,29]. In the case of a prosecutor attack where the adversary has a priori information about the person, linking databases can give extra information for the adversary. The main difference between the medical and the legal databases is that the database contains not only the data fields, but the context of the text can contain many quasi-identifiers, which can help the attacker gain some new information. For example, suppose the attacker wants to know for whom the orthopedist performed the surgery on the wrong side. In that case, he would only check the local journals and the homepage of the small hospital, and he will know not only the name of the doctor but also the patient’s information from the text of the legal document. The connecting cases can link other persons to this document. The previously mentioned algorithms, which were developed for medical data anonymization (k-anonymity, t-closeness, and l-diversity) can help to reduce the number of the quasi-identifiers in the text. However, the over-usage of these algorithms can significantly reduce the understandability of the text. Moreover, the quasi-identifier structure leads to a sparse matrix, not a dense as in medical text anonymization, where the methodologies mentioned above developed. Due to the asymmetries between these databases, the mathematical model of these two fields leads to different approaches.

Aa an example of a journalist attack, we can examine the case of a small district court, where there are a small number of potential criminals. If we know some marker of a criminal or a criminal group, which committed many similar cases in the area, we can connect the involved persons via the connecting legal documents. This can easily lead to a lot more sensitive information about the involved people.

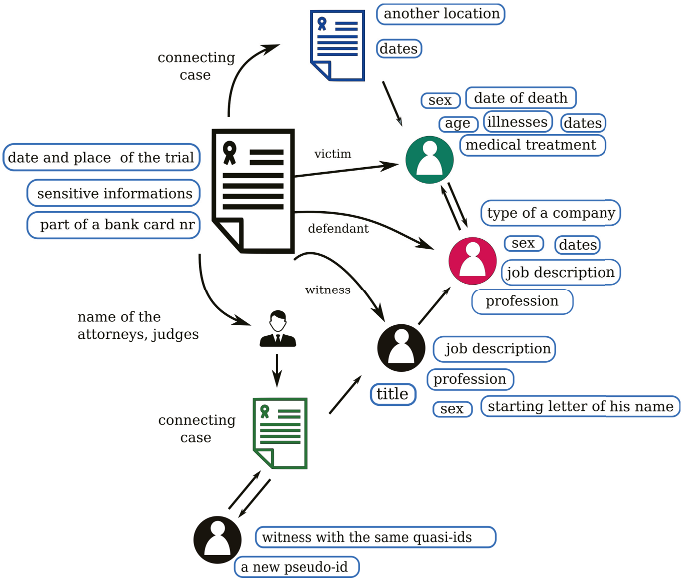


Figure 4. The image shows how a published legal document can connect to other legal documents directly or via the judges and attorneys, and the connections between the recognizable quasi-identifiers, which can be assigned for different people. These identifiers can be used to perform a linking attack, linking data to other databases.

4. Does Document Domain Matter? Differences between Medical and Legal Anonymization Tasks

At first sight, medical data and legal texts share several similarities (domain-specific language, structured content to some extent, a wide range of data types to be anonymized etc.), there are specific problems that characterize legal documents only. Medical data is often available in unstructured text format (e.g., clinical notes, radiology, and pathology reports, and discharge summaries) [66] where legal cases only exist in this form.

One main difference would be that in the majority of medical text there are at most two subjects mentioned: one is the patient, who has diseases, several IDs, job, and the other is the hospital where he or she is being treated. It is relatively rare to find data that is pointing out from this context (e.g., other people, natural formations, car plate numbers etc.). Therefore, linking the extracted entities to a specific person is obvious in medical texts [67]. Whereas in the majority of legal cases there are also two parties (plaintiff and defendant), it is not rare to have more than one person on any or even both sides. Not to mention the witnesses, companies involved, bank accounts, etc., are also mentioned in the texts.

Another significant difference is that due to the nature of legal cases, the matter of fact describes the whole story of the parties in detail and usually this part of the legal decision is full of complex quasi-identifiers. These complex quasi-identifiers mean events or rather chains of events that could be easily used by an attacker to form further queries while attempting to link data sources to the parties of a legal decision. An example would be an extraordinary or rarely occurring event, e.g., someone was gored by a bull, died during a gland surgery or breeds limousine-type cows that could easily appear in local newspapers or publicly available databases. It is important to point out that how much a rarely occurring event makes a difference, since that type of data does not appear in tabular health data, whereas the legal documents contain many of these, especially the matter of fact part. Moreover, the definition of "personal data" in GDPR Article 4 states that these data can be considered as indirect identifiers (Regulation (EU) 2016/679 of the European Parliament and of the Council Article 4).

Emphasizing the role and importance of quasi-identifiers in anonymization is not only the result of GDPR. About 60 years ago researchers started investigating the idea whether a small number of data points about an individual can be collectively equivalent to a unique identifier even if none of these data points are unique identifiers [33,73–75].

Health data of patients are often presented in a tabular format and these data tend to share the same columns (i.e., a traditional relational database). In this dataset, the rows (records) represent data of a patient and the columns represent the attributes a patient has (e.g., sex, date of birth, profession, disease, etc.). These attributes usually do not contain many null elements, every attribute, every column are well represented. This is not the case in legal cases. If we represent the data stored in a set of cases the same way as one does with medical records, the number of attributes, i.e., columns, can be a great deal more than in a medical database, so the dimension of the records is higher. Moreover, these data may appear relatively rare in other records, hence the matrix that could be created from a set of cases is sparse (see Figure 5). The occurrence of data that fall under the regulation of GDPR is highly asymmetric in legal documents. This sparsity puts the complexity of the problem to a completely different level compared to medical records.

The right side of Figure 5 illustrates the case of a typical medical database, where every row contains the same information about an individual. This database is symmetrical because every record has the same identifiers, and we know every possible quasi-identifier in this task. The health data anonymization methodologies use this symmetry or structural regularity. In contrast, the legal documents can be very asymmetrical, hard-to-find similar structural regularities, which increases the complexity of the risk analysis of these models. The most obvious example is the role of the personal data of the dead persons in the documents. This data belongs to the GDPR regulations in some countries, but some countries do not care about the personal rights of dead persons. However, in probate proceedings, some sensitive data of the dead person can be published. If we know the link between the dead person and the plaintiff, who is, e.g., the only living relative, we can re-identify their data. A more general risk for the asymmetrical dataset is that the document contains three-three quasi-identifiers for two individuals, which is insufficient to re-identify the involved persons. However, these six identifiers can belong to six different categories (such as occupation, age, etc.), and by knowing the relation between the individuals, it can be possible to re-identify them.

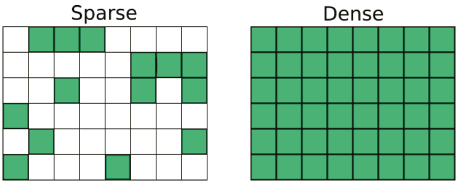


Figure 5. Sparse (asymmetric) and dense (symmetric) representation of data.

There is a theoretical proof for the statement that high dimensional data is vulnerable to de-anonymization [33,34,49].

Narayanan and Shmatikov claim that “Most real-world datasets containing individual transactions, preferences, and so on are sparse. Sparsity increases the probability that de-anonymization succeeds, decreases the amount of auxiliary information needed, and improves robustness to both perturbation in the data and mistakes in the auxiliary information” [21].

Thus, the task of anonymization in legal documents cannot, by its very nature, be regarded as being entirely solvable.

A good example for de-anonymizing sparse data would be the case study about the Netflix Prize dataset [20,21,41] and the America Online (AOL) search engine query log dataset [42].

Motwani and Nabar published an algorithm that achieves k-anonymity on an unstructured, non-relational dataset, namely on search engine query logs [24]. In this dataset, there was no need to distinguish between direct and quasi-identifiers. Their approach was to transform the tokens from the query logs to a relational database containing only ones and zeroes. This database had a high number of dimensions and was sparse. They then reached k-anonymity by adding and deleting as few elements as possible from each query until the k-anonymity criterion have been met. Although, in many aspects of their data, the problem itself is different from the data that could be retrieved from legal documents, the solution can be useful in reaching k-anonymity in legal documents as well.

As a consequence, performing decent anonymization in legal cases is far away from just identifying direct identifiers from the text and deleting or modifying them; in other words, to only perform Named Entity Recognition and modify the extracted entities. Nevertheless, the currently available anonymization software solutions generally represent this mindset [2,9,25].

To perform a decent anonymization on a legal document, a wide range of quasi and direct identifiers have to be recognized, in particular, the rare events mentioned in the case. After the recognition and modification of direct identifiers, the quasi-identifiers have to undergo a careful risk analysis, i.e., the risk for co-occurrence of many quasi-identifiers that can be connected to individuals has to be estimated and if needed, anonymization techniques have to be applied on them. According to our knowledge, there is currently no anonymization solution for legal texts that takes the importance of events into consideration [2,7,9,58,76,77].

Data owners have to accept that legal cases may not be de-identified [33,34,49], but not protecting these data at all is also not an acceptable option. The problem is similar to having our bicycle protected from being stolen. It is not a good tactic to park somewhere and hope for the best. Even though there is no bike lock that cannot be broken, it is still worth using at least some bike lock, because many times that is enough to deter the thief. Instead of giving up anonymization, data owners should aim to reduce the chances of an attack succeeding.

5. Structure and Privacy Risks in Hungarian Legal Documents

In this section, we provide an overview of the current legal system and known regulations related to anonymization in Hungary focusing on the risks that legal documents inherently contain. For other member states in the European Union, the work of van Opijnen et al., [7] provides a good general overview of data protection in the legal domain.

5.1. Judicial System, Regulations

Hungary has a four-tier judicial system, which consist of the district courts, the regional courts, regional courts and the Supreme Court [7].

Organisation and Administration of the Courts regulates the publication of legal documents and decisions by the Act CLXI of 2011 in Section 5.1 (“Responsibilities of Courts Relating to the Publication of Court Decisions; the Register of Court Decisions”) and articles 163–165 [7].

Article 164 regulates that after the decision is rendered in writing, it shall be published by the chairman of the court in the Register of Court Decisions within thirty days (Act CLXI of 2011 on the Organisation and Administration of the Courts Article 164).

The law currently in force provides two different forms of publication. In the first case, the decisions have to be published automatically, but in the second case, publication depends on the will of the litigants. In both cases, the published documents have to be anonymized. Therefore, not all decisions are subject to the obligation to publish; decisions of lower courts are outside the scope, if the legal procedure does not reach at least the regional courts of appeals.

According to current regulations, e.g., the name of the court concerned, the name of involved judges, lawyer acting as an agent, the defense counsel or administrative organizations (e.g., National Tax and Customs Administration), or the authors of certain scientific publications are to be anonymized [7,8]. Sensitive data have to be deleted in a way that the deletion does not change the established facts of the case. Instead of the names of the persons covered by the decision, the name corresponding to their role in the procedure shall be used; instead of the names used to identify the persons and the protected data, the name of the data type shall be used as a replacement text [8], e.g., if there are "i" number of plaintiffs mentioned in the decision it should be replaced as $Plaintiff_1$,$Plaintiff_2$,...$Plaintiff_i$.

The law also states that all data enabling the identification of a natural person, a legal person or an organization without legal personality have to be removed. Nevertheless, the exact range of data to be anonymized is not defined.

Recital 27 of GDPR (gdpr-info.eu/recitals/no-27/) (accessed on 1 May 2021) states that the Regulation does not apply to the personal data of deceased people and leaves this question to the EU Member States. In Hungary, the Act CXII of 2011 (InfoAct) states that the data subject's rights after their death could be exercised either by a person appointed by the data subject during their life or a close relative (www.twobirds.com/en/in-focus/general-data-protection-regulation/gdpr-tracker/deceased-persons) (accessed on 1 May 2021).

5.2. Criticism of Current Regulation

Although the current regulation tries to embrace the rules defined by the GDPR, some points can be a subject of debate.

First of all, it is important to point out that the Hungarian law does not distinguish direct and quasi-identifiers. This is problematic, since one could easily associate only with direct identifiers and we will see that this is the case when it comes to practice. With quasi-identifiers remaining in texts, the parties could become easily identifiable [40].

It is important to remark that mentioning a rare event has to be considered as a quasi-identifier (sometimes as a direct identifier), but removing or modifying this can easily contradict the commandment stating that the sensitive data have to be modified in a way that they do not change the established facts of the case [7,8]. Nevertheless, in these cases, generalization of the events could be a possible solution for the contradiction.

Another remark about the law would be that it requires the decisions to be published within 30 days. Although the year when a case started can currently be determined from the case identifier and the end of the case is also mentioned in the document, the results on the Netflix dataset may serve as a basis for revising this condition [20,21].

5.3. Current Practice and Potential Risks

The practice of anonymization on court decisions shows that the names of the parties and all their data (address, mother's name, date of birth etc.), so the direct identifiers have been completely removed or in certain cases people's full names or company names have been replaced with their monograms.

Consequently, the texts usually contain a remarkable amount of quasi-identifiers. Generally, every attribute that can be used to perform a linking attack, is a quasi-identifier. Some examples would be: sex, age, locations, professions, monograms, dates (day, month and year), company types, activities of a company. The name of judges or attorneys involved could provide further links, e.g., the name of the lawyer, even if all data related to location has been removed/replaced in the document, could provide additional information on the location of the events, especially if the lawyer is placed in the rural area.

5.4. Case Studies

In order to present the power of quasi-identifiers and also to identify some of the potential quasi-identifiers in the legal domain, research has been conducted using publicly available legal cases in Hungarian.

5.4.1. Datasets and Search Framework

Our case study was conducted by using the LEXPERT service (https://www.lexpert.hu) (accessed on 1 May 2021), which is a self-developed online database of the freely available Hungarian legal cases. LEXPERT ensures an Elasticsearch-based search platform for the legal documents and links the connecting documents and law references. The following documents are included in this database:

- Case-law of the Constitutional Court: decisions and procedural decisions of the Constitutional Court of Hungary, currently approximately 9800 documents.
- Court Decisions: consists of decisions from all tiers of the Hungarian judicial system from district courts up to the Supreme Court. This is the biggest dataset containing approximately 160,000 documents. This study was conducted using this dataset.
- Uniformity Decisions: binding decisions of the Supreme Court in uniformity cases to ensure the uniform application of law within the Hungarian judiciary, currently containing approximately 200 documents.
- Selected case-law: selection of the most important court decisions, currently containing approximately 2200.
- Division opinions and decisions: decisions of the divisions of the courts on abstract, theoretical legal questions. Contains approximately 400 documents.

5.4.2. Illustrative Examples

One of the simplest cases was when a judge started a labour lawsuit against their employee. Since names of attorneys and judges do not have to be removed, in this case, their workplace and their name remained in the text.

We have noticed that many documents contain exact dates, which is a possible source of a serious information leak. Sometimes, not only the year, month and day are mentioned but the hour and minute as well. When these dates refer to the general flow of a legal case (e.g., dates of previous decisions, appeals etc.) it usually does not mean a potential risk.

However, when a date refers to a unique event (e.g., someone was gored by a bull, died during a routine surgery, is starting a business, etc.), it can pose a serious threat for re-identification, since these data usually appear in local newspapers or can be looked up in the Hungarian Company database or other publicly available databases. The death date is handled differently in the different EU member states, for example, in Hungary it is not personal data; however, in Denmark it remains personal data after the death of the person [78].

An example for the possible threat that multiple quasi-identifiers may pose would be the following case. In southern Hungary, there was a case when a then-92-year-old deceased person's date of death and age were published. The authors of [79] showed that this is sensitive information, because this age can be applied to no more than the 0.5 percent of the population. It is known that, at that time, around 4000 women lived in Békés county with over 85 years of age, from the total population of 397,791 [80]. This information cannot be sensitive because this represents one percent of the total population, and if we did not know the sex of the victim, the age could refer to nearly the 1.5% of the population. After a subtraction, we know exactly her age when she died and there were no more than 300 people. Of these, 200 were woman in that county and were 92 years old; this is the 0.05% of the total population, which is significantly lower than the recommended minimum (0.5%). With only two pseudo-identifiers, we were able to narrow down the number of potential individuals to around 200. In this case, we did not even use the other pseudo-identifiers that could be learnt from the text, the exact date of the death and where she lived. From the text, we know that the woman lived in a small town with no more than 5000 inhabitants. This means if the old people are distributed uniformly in the examined area, that there were four different people at that time fulfilling these criteria. The shrinkage of equivalence class sizes can be followed in Figure 6. This information can be paired with the local journal's obituary section, easily identifying the dead person.

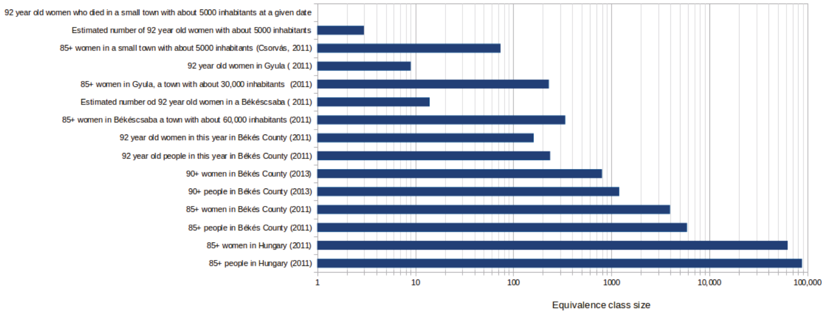


Figure 6. The image shows the risk if the age and the gender of an old person (85+) is published in a legal document. The different bar plots shows the equivalence class sizes, when the document does not contain or contains information about settlement involved [80].

Naturally, the de-anonymized data could be used for linking entities further.

Another potential risk factor is that the documents related to the medical field often publish the full medical history of a person with exact dates and types of surgeries, drugs taken, etc. By themselves, these data could be used to reveal someone's identity; however, linking these to public databases can be difficult. The problem is that this information can be sensitive. We have found a case where only the case of medical malpractice was enough to identify both parties with a simple Google query. It made the matching process easier that the dates were also mentioned in the document. In this example, the whole medical history of the patient was mentioned referring to sensitive types of surgeries and medical treatments.

Another date that appears in each document is connected to geolocation, namely, the courts involved. Since the population density is not equally distributed across the country (which is likely the case for the majority of the countries in the world), a court operating in a smaller county can be considered as a quasi-identifier. For example, if there is a hospital involved in a case and the type of surgery is mentioned, in many cases, the name of the institute can be easily re-identified. If any name of a settlement accidentally remains in the text, the re-identification can be even easier. For example, a particular issue in the Hungarian language is that when a name of a settlement is referred as "something from Xy settlement", the name of that settlement is not written by capital letters but lower-case letters. The human annotators tended to miss these data.

Another quasi-identifier would be when a natural or artificial formation type (e.g., reservoir, lake, cave, river, mine etc.) is mentioned in the text even if the name of it is completely removed. Since the region of the case is known from the court, these formations can be unique identifiers. There was a case where the starting characters of the settlements involved remained in the text, alongside the term reservoir of the river XY. Since there is only one of this, the settlements could easily be identified. The text also contained fragments from parcel numbers, thus the owners could be identified from these data.

The profession could be another quasi-identifier that can seriously reduce the equivalence class size, hence increasing the risk of de-anonymization. For example, there was a case in which academic members were involved. In itself, this information reduces the equivalence class size to around 300. The case mentioned the scientific domain alongside the monograms of these people, giving more than enough information to de-anonymize the parties.

In case of companies, the scope of activities could also be a potential identifier. Even if this information would not identify a company uniquely, knowing the date when it started and in which city would provide enough information to do so, as we found in another criminal case. Since the members of one company can be listed even historically in the Hungarian Company Database, it is relatively easy to find the accused people. In this case, many fragments of bank account numbers were also available alongside the name of the banks, providing additional information.

Using a good name entity recognition together with a non-appropriate anonymization technique can be a double-shaped blade. There was an example case in the published data, where a masking technique anonymized the hospital's name as "U... Rt.". From this masked name, we have three pieces of information. First, the presence of the "..." refers to the fact that these data are a direct identifier. Secondly, the medical treatment was made in a private hospital because "Rt" means public limited company, and we know the company's starting letter. There were only two different medical service companies in Hungary at that time, which name starts with U, but only one of them is authorized to perform such medical treatment. However, if the name of the hospital had only been generalized or suppressed perfectly, we could have not looked this information up in the Hungarian Company database.

We have provided some examples of how only the company type, the location of the company, and the date of registration can reduce the equivalence class sizes. The results can be seen in Figure 7. Knowing these three pieces of information reduces the equivalence class sizes so drastically that only Budapest could make this value above 10,000 equivalence class size compared to the whole country. Even in the case of the cities with the largest population after Budapest (Debrecen, Miskolc, Szeged, Győr), the size of the equivalence class reached, at most, 400.

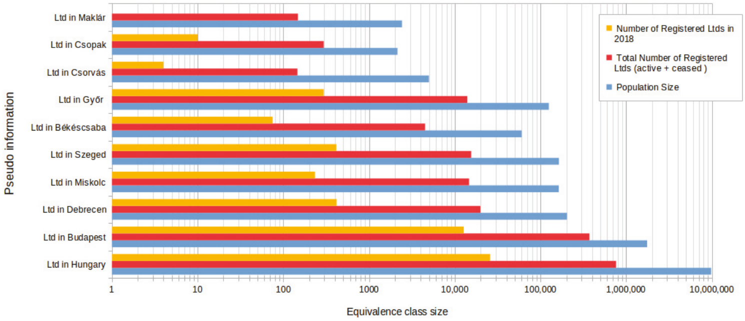


Figure 7. The image shows the risk if the company form is published together with the date of the registration, in the case if the headquarters is known or unknown from the corpus. The blue bar shows the total number of the registered active or inactive Ltds in Hungary or in the given city. The orange bar plots those Ltds, which are registered in 2018 [80].

To sum up the results, it can be stated that the type of quasi-identifiers is broad not only in domain (from dates to natural formation types) but also in their nature (simple nouns to chain of events).

The risk of re-identification dramatically increases when the case can be connected to time and location, and all documents are connected to a court, and it has been year at least since the case started, as these data must be published according to the current regulation. Generally, following common sense, all data, which is rare in some sense, increases this risk. The cases highlighted in this section showed that mentioning exact dates and starting letters should be avoided, since these additional pieces of information reduce equivalence class sizes drastically. Nevertheless, the risk of re-identification should be estimated by involving as many quasi-identifiers as possible from the given text since considering these data together, there may already be enough information for de-anonymization. The question arises: how could be this risk quantified? We are pursuing the answer for this question in the following section.

5.5. Quantifying Risk

Intuitively, it can be stated that not all quasi-identifiers have the same "strength", "danger-factor" or the same information content as far as de-anonymization is concerned.

In Information Theory, the metric for measuring information is entropy. Shannon Entropy has been defined in Definition 9.

Theorem 9 (Shannon Entropy [81])**.**

$$H_S = -\sum_{n=1}^{N} p_i \cdot \log_2 p_i \tag{2}$$

where P = ($p_1, p_2, ...p_N$) is a discrete probability distribution and $p_i \geq 0$ and $\sum_{n=1}^{N} p_i = 1$

Entropy is measured in bits. Assuming that we have an equivalence class with size N and all members of this class occur with the same probability, by picking randomly, on average, $log_2 N$ guesses have to be made to find a certain element from that equivalence class and $log_2 N$ is equal to the Shannon Entropy. If the entities from an equivalence class do not have the same probability for being the element to be found, the entropy will be lower; in case when it is known, $H = 0$ stands. This means that a theoretical maximum entropy can be calculated, assuming that the attacker is performing random picking on the equivalence class.

The size of an equivalence class could be gathered from demographic statistics or other publicly available data sources such as company databases, medical databases, voters registration list and so forth.

In case the exact size of the equivalence class characterized by the extracted set of quasi-identifiers is not known, by calculating conditional probabilities or applying Bayes' Theorem [82] it is possible to estimate the size of this equivalence class.

Theorem 10 (Conditional probability [83])**.**

$$P(A/B) = \frac{P(A \cap B)}{P(B)} \tag{3}$$

Theorem 11 (Bayes' Theorem [82,83])**.**

$$P(A/B) = \frac{P(A) \cdot P(B/A)}{P(B)} \tag{4}$$

However, the easiest solution would be to assume that $A_1, A_2, ...A_i$ probability variables are independent to each other, and the conditional probabilities of these attributes can be calculated as: $P(A_1 \cap A_2 \cap ... \cap A_i) = P(A_1)P(A_2)...P(A_i)$. Although the independence is not true in many cases, this simple equation could be easily used as an estimation in many cases.

Hence, by extracting and linking attributes to a specific person and estimating the equivalence class size of all these attributes occurring together, the entropy can be estimated as well. However, it is important to point out that by performing the estimation using more parameters, the level of potential error also increases, and with it the variance of potential information content also increases. For instance, in two cases having equal equivalence class sizes but one has been estimated using two attributes and the other has been estimated using four, it is expected that the latter scenario can be riskier in terms of de-anonymization.

Figure 8 shows the connection between entropy and risk.

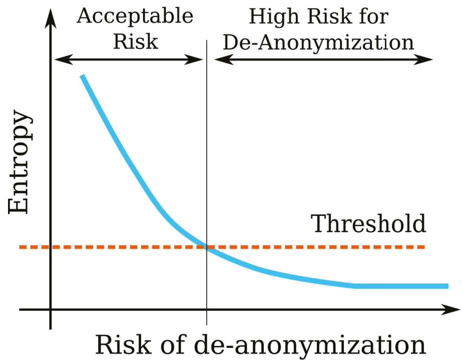


Figure 8. Connection between entropy and risk.

According to our understanding, the risk as function of entropy is such that by decreasing entropy (by knowing more of a specific person) the risk increases.

Since there is no case when the probability of de-anonymization is zero, the de-identification process must aim to raise the entropy of the set of quasi-identifiers above a certain threshold.

5.6. The Threshold

As Figure 8 suggests, there is a threshold dividing the acceptable and high-risk regions. The question is, then, what is a reasonable choice as a threshold value? The answer is, it depends on the actual application. The authors of [34] suggested a weighting for attributes based on the count of non-zero elements of an attribute and define a condition for attribute sparsity (Definition 13). Since, in our case, the real size of the equivalence class usually cannot be determined using the set of legal documents but requires other publicly available databases, the definitions have been modified accordingly.

Theorem 12 (Weight of an attribute [34])**.** *"A weight of an attribute is $w_i = \frac{1}{log_2 N}$ where N is the size of the equivalence class".*

Theorem 13 (t-sparsity [34])**.** *"An attribute is t-rare if $w_i = \frac{1}{log_2 N} \geq t$ where N is the size of the equivalence class and t is a threshold value, $0 < t \leq 1$".*

In the Netflix case study in [34], $t = 0.07$ and $t = 0.075$ have been used as sparsity values, suggesting approx. 20,000 and 10,000 equivalence class sizes, 13.33 and 14.29 bits of entropy, respectively.

In [79], a rule of thumb has been mentioned in terms or equivalence class sizes: 0.5% of the population that would mean 48,850 in Hungary (approx. 15.6 bit or entropy). The presentation has mentioned examples, such as settlements under 10,000 population or the population above the age of 85 being considered dangerous, hence an anonymization method (generalization or suppression) has to be applied in these cases.

To provide some examples, entropy values and weights for attributes have been calculated and presented in Table 1. Surprisingly, 33 bits of information are enough to identify someone uniquely from the world's population [33].

Table 1. "33 bits of entropy are sufficient to identify an individual uniquely among the world's population" [33].

	H (bit)	**w (1/bit)**
World's population	32.86	0.03
Hungary's population	23.22	0.04
Budapest's population	20.74	0.05
Age 85+	17.61	0.06
Medical doctors	15.33	0.07
Professional football player	10.19	0.10
Nr. of companies	20.28	0.05
Nr. of Ltds	19.54	0.05
Nr. of Ltds founded in 2018	13.63	0.07
Nr. of Ltds founded in Jan 2018	10.26	0.10
Hungarian Academy of Sciences (HAS) member	8.15	0.12
Member of HAS in Engineering	4.91	0.20

From the point of view of the data owner, it is crucial to know or at least to estimate the information gain when an additional attribute is known. This information gain appears on the attacker's side.

Theorem 14 (Information gain [83])**.**

$$IG(Y/X) = H(Y) - H(Y/X) \tag{5}$$

where IG is the information gain on the event Y if X is given, H(Y) is the entropy of event Y and H(Y/X) is the conditional entropy of the event Y given event X.

Theorem 15 (Conditional entropy [83])**.**

$$H(Y/X) = \sum_{x \in X, y \in Y} p(x,y) \cdot log(\frac{p(x,y)}{p(x)}) \tag{6}$$

alternatively,

$$H(Y/X) = \sum_{j} P(X = v) \cdot H(Y|X = v) \tag{7}$$

where H(Y/X) is the conditional entropy of the event Y given event X, P(X = v) is the probability of event X taking the value v, H(Y | X = v) is the entropy of event Y if event X takes the value v.

Denote by X and Y that there is an additional attribute and there is a person in the legal text. In legal texts, usually, not only the attribute X is mentioned, but so is a certain value of it, denoted by v. If it is possible, $H(Y/X = v)$ has to be calculated to see how the additional information decreases the complexity of the given problem. In case it is not possible, the entropy of event X = v can be estimated by considering event X generally, since $H(Y/X = v) \leq H(Y/X)$. In a worst-case scenario for the attacker, if we consider the probability distribution of X as uniform, the conditional entropy $H(Y/X)$ will have the highest value of all other distributions. It can be seen that from the data owner's point of view this is the best-case scenario so this is a weak estimation. Despite that, this estimation still can be useful, since, if in the worst-case scenario for the attacker, the entropy of the problem decreases under the threshold value, it can be stated that an attacker is likely to have enough information to re-identify data. Another possibility for estimation would be, when the probability distribution of event X is not uniform, but the most probable values are known with their probabilities. In this case, the minimum entropy would be $H_{min}(X) = -p_i \cdot log_2 p_i$ where i denotes the most probable element of the distribution. It is clear that in this case $H_{min}(X) \leq H(X)$.

In Table 2, some examples have been provided to show how much a given problem could be simplified by a piece of additional information. The calculations assume that both the problem and the additional information (e.g., the monograms) are uniformly distributed. Despite the fact that the Hungarian alphabet is 44 letters, we took that into account with only 40 letters during the calculations, not counting the letters q, w, x, y, which are extremely rare in the Hungarian language as starting letters. During the calculations, the full English alphabet was taken into account.

If the entropy of a problem is known and it can be estimated that how much a piece of additional information reduces the fraction of the equivalence class size, the entropies can be just subtracted from each other since $log_a(b/c) = log_ab - log_ac$.

If there is a case mentioning the two-letter monogram of a witness and provides the additional information that he/she is an academic member, it is highly probable that the person can be identified, since knowing that someone is an academic member means 8.15 bits of entropy and subtracting 10.64 bits for knowing the monogram would result in negative entropy, and zero entropy would mean that the person has been identified. When we consider the difference between the entropy of Nr. of Ltds founded in 2018 and Nr. of Ltds founded in January 2018 from Table 1, we can learn that the difference is 3.37 bits, which is close to the value of 3.58 bits potential information gain presented in Table 2.

Table 2. Possible information gains when additional information is known.

	Information Gain
Sex	1
Month, when year given	3.58
1 year from 30 years range	4.91
Year and month in 30 years range	8.49
Year, month day from 30 years range	13.42
Monogram Hungarian (1 letter)	5.32
Monogram Hungarian (2 letters)	10.64
Monogram Hungarian (3 letters)	15.97
Monogram English (1 letter)	4.70
Monogram English (2 letters)	9.40
Monogram English (3 letters)	14.10

6. Automatized Workflows for Pseudonymization

Using an automatized named entity recognition workflow is not enough to comply with the GDPR. The importance of different pseudo identifiers must be considered during the anonymization workflow. The available automatized pseudonymization applications should be improved using different methodologies such as event recognition, semantic role labeling and risk analysis to create effective pseudonymization tools (Figure 9). In this section, the parts of the process are described.

Given a legal text, finding direct and as many quasi-identifiers as possible is the first step towards the pseudonymized legal document. This is carried out by recognizing these entities from the text. This process is also known as Named Entity Recognition (NER) [84–86]. There is a vast number of anonymizer solutions that are based on NER models [2,25,58,66,67,87–89]. As a consequence, the performance of the NER model used in any anonymization architecture highly influences the performance of the anonymization solution [25]. However, it is important to note here that finding direct identifiers is a necessary but not a sufficient condition for anonymization [14–16]. Finding pseudo-identifiers is also important, but the range of quasi-identifiers can be wide due to the nature of legal texts. This is because a part of a legal case, namely the matter of fact, describes the whole story of the parties in detail, and, usually, this part is full of quasi-identifiers that are hard to discover automatically. In practice, finding all types of quasi-identifiers is a very difficult task and the selection of quasi-identifiers should be preceded by a careful risk analysis.

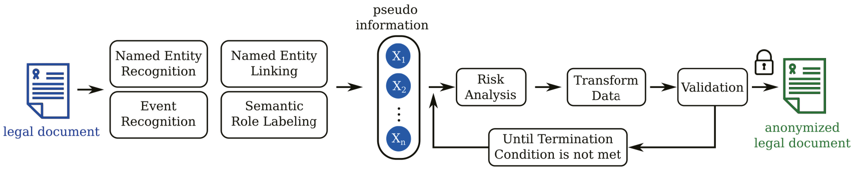


Figure 9. Named Entity Recognition should be extended with Named Entity Linking, Event Recognition and Semantic Role Labeling to realize a GDPR compatible pseudonymization framework for legal cases.

In contrast to medical records, at this stage, the extracted data cannot be put into tabular format to perform risk analysis. This is because medical health records generally contain information about one person but this is not the case in a legal document. Usually, there are two sides (plaintiff and defendant) and there could be more people involved on each side. The case could also mention other people involved indirectly, e.g., witnesses, doctors, visited places.

To solve this issue, the extracted name-typed named entities (e.g., name of person or institute) have to be connected to the other extracted entities (dates, age, profession, etc.). One possible solution to perform such identification of connections would be via dependency parsing. Once entities have been extracted, wherever possible, an equivalence class size has to be assigned to each entity, either by using a knowledge base or by looking up tables of statistics. This part of the process is denominated as Named Entity Linking [90] in Figure 9.

Since the importance of events as quasi-identifiers has been emphasized in this study, the events have to be extracted from the texts alongside the arguments of these events. This process is known as Event Extraction [76,77,91–93].

The task of specifying "who did what to whom where and why" from text is called Semantic Role Labeling [94]. It is important to point out that from the anonymization point of view, classification of events based on their rarity is more important than finding all the answers to the questions mentioned above. Nevertheless, these answers could lead to a better rarity estimation.

As a result of this stage, all extracted information can be stored in a matrix where each row refers to a name-typed entity and each column to a specific attribute of this entity. Motwani and Nabar have shown that it is possible to transform unstructured data into a relational database format containing only zeroes and ones, which is sparse [24].

Once these data have been collected, the risk analysis is the next task. By risk analysis, we mean estimation of equivalence class sizes of each attribute connected to a specific named entity (e.g., Person type) by using knowledge bases and/or demographic statistics, or third-party databases. In the case of events, the focus is on estimating rarity similarly to the non-event type entities. From these data, the risk could be estimated by calculating entropy values for each data extracted from the text and, more importantly, to the collection of these data and comparing them to a given threshold.

The next step is transforming the data. This step is the application of anonymization techniques such as generalization, masking, slicing, suppressing, etc., on the extracted data. By these techniques, the risk of re-identification is decreased. The modified entities then have to be put into the document, replacing the extracted forms.

As a final step, the validation of the anonymized document is made from two aspects. On one hand, the likeliness of re-identification and, on the other hand, the intelligibility of the document. This is important in legal decisions, since, for example, if every entity were replaced by "..." characters, the text would become confusing due to the relatively high number of participants in a legal case.

Should the anonymized text fail on validation, the whole procedure is repeated from the risk analysis step until the termination condition is met.

About the Feasibility of a GDPR Compatible Automatized Pseudonymization Framework

By using the technologies suggested in Section 6, the chances of identifying the people involved in a case can be significantly reduced. To achieve a GDPR-compatible anonymization process, which means that all of the involved people have been de-identified, each step (Named Entity Recognition, Event Recognition, Semantic Role Labeling, etc.) must work with very high efficiency. We can think it over in the following example, which considers only the importance of the NER from the tools as mentioned earlier. If one has a NER model that recognizes the sensitive entities with 99% accuracy, and there are 20 entities in the text, which should be identified, the probability that at least one entity (quasi-identifier) will remain in the document is still significant: $1 - 0.99^{20} = 18.2\%$. This is why one of the pillars of the TILD methodology [27] is to test the anonymization systems via motivated intruder testing involving humans [95]. Most of the state-of-the-art pseudonymization tools use the NER to create a pseudonymized document only. This does not ensure that some person can be re-identified after some specific event. On the other hand, if all of these entities are replaced, these entities should be replaced with some other words that fulfill the grammatical role of the original text in order to preserve the information in the content and the clarity of the text. This justifies that a good semantical analysis can improve the quality of the pseudonymization process.

The conclusion is that data owners have to accept that legal cases may not be fully anonymized, only pseudonymized with an acceptable risk [33,34,49]. Data owners can reduce the chance of a successful attack by conducting a risk analysis and applying the proposed pseudonymization technologies presented in this section. The existing risk analysis methodologies are based on databases, where the distribution of the information is symmetrical, such as medical databases, where every record has the same properties. Legal documents are unstructured sources of possible quasi-identifiers. The database built from the linking documents is a large, asymmetrical dataset, which should be considered to create different and more effective risk analysis and pseudonymization algorithms.

7. Conclusions

The digitalized, openly accessible court decisions have a fundamental role in improving the decision-making processes and making the administration of judicial systems more transparent. The GDPR forms strict regulations for openly publishing private data. Therefore, the published data of the mentioned parties of the court decisions should be pseudonymized. During the pseudonymization process, the direct and indirect identifiers are masked, generalized, or replaced by taxonomies. The main difference between anonymization and pseudonymization is that the latter process is reversible. However, anonymizing the data can often destroy the utility of the published data.

Many automatized solutions have been developed in the different EU member states to accelerate the solution of this process. Most of these tools use modern named entity recognition-based methods to classify, mask or generalize the direct identifiers. Therefore, the solutions mentioned above pseudonymize these documents only. Nevertheless, these tools do not take risks and the utilizability of the pseudonymized data into consideration. The legal documents are unstructured texts where the information-loss after removing the different parts of the sentence should be considered. Privacy-preserving publishing can be achieved by the application of differential privacy algorithms, as in the case of public health data. However, the structure and the information content of a legal case greatly differs from health records, where the same type of data represents every individual. In legal documents, a wide range of attributes of different kind are available, referring to the involved parties. Moreover, legal documents can contain additional information about the relations of the involved parties and rare events. Hence, the personal data can be represented by a sparse matrix of the attributes. It has been shown that this kind of anonymized data is inherently prone to de-anonymization.

Therefore, a named entity recognizer tool is essential to make a fair anonymization process, but it is not enough. Named entity recognition, event recognition, semantic role linking, named entity linking should be used together with the anonymization algorithms (k-anonymity, l-diversity, etc.) to quantify the level and the utility of the process. The risk analysis can be performed by using statistical methods and entropy by estimating equivalence class sizes. To sum it up, due to the No Free Lunch Theorem [68–70], there is no single easy solution that exists for anonymization that works for all approaches in all possible scenarios.

Author Contributions: Conceptualization, T.O.; methodology, G.M.C. and T.O.; software, G.M.C., D.N., J.P.V. and R.V.; resources, D.N. and J.P.V.; writing—original draft preparation, G.M.C., R.V., D.N. and T.O.; writing—review and editing, D.N. and T.O.; visualization, G.M.C. and T.O.; supervision, T.O.; project administration, D.N.; funding acquisition, J.P.V. All authors have read and agreed to the published version of the manuscript.

Funding: Project No. 2020-1.1.2-PIACI-KFI-2020-00049 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2020-1.1.2-PIACI KFI funding scheme.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Comission, E. Digitalisation of Justice. Available online: https://ec.europa.eu/info/policies/justice-and-fundamental-rights/digitalisation-justice_en (accessed on 1 June 2021).
2. Oksanen, A.; Tamper, M.; Tuominen, J.; Hietanen, A.; Hyvönen, E. ANOPPI: A Pseudonymization Service for Finnish Court Documents. In Proceedings of the Legal Knowledge and Information Systems JURIX 2019, Groningen, The Netherlands, 12–14 December 2019; pp. 251–254.
3. Velicogna, M. In Search of Smartness: The EU e-Justice Challenge. *Informatics* **2017**, *4*, 38. [CrossRef]
4. Hyvönen, E.; Tamper, M.; Ikkala, E.; Sarsa, S.; Oksanen, A.; Tuominen, J.; Hietanen, A. LawSampo: A semantic portal on a linked open data service for Finnish legislation and case law. In Proceedings of the ESWC, Heraklion, Greece, 31 May–4 June 2020.
5. Oksanen, A.; Tuominen, J.; Mäkelä, E.; Tamper, M.; Hietanen, A.; Hyvönen, E. Semantic Finlex: Transforming, publishing, and using Finnish legislation and case law as linked open data on the web. *Knowl. Law Big Data Age* **2019**, *317*, 212–228.
6. Csányi, G.; Orosz, T. Comparison of data augmentation methods for legal document classification. *Acta Technica Jaurinensis* **2021**. [CrossRef]
7. Van Opijnen, M.; Peruginelli, G.; Kefali, E.; Palmirani, M. On-Line Publication of Court Decisions in the eu: Report of the Policy Group of the Project 'Building on the European Case Law Identifier'. SSRN 3088495. 2017. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3088495 (accessed on 1 June 2021).
8. Országos Bírósági Hivatal (National Office for the Judiciary). 26/2019. (XI. 25.) OBH Utasítás. 2019. Available online: https://birosag.hu/obh/szabalyzat/262019-xi-25-obh-utasitas-birosagi-hatarozatok-anonimizalasaval-es-kozzetetelevel (accessed on 7 August 2021).
9. Povlsen, C.; Jongejan, B.; Hansen, D.H.; Simonsen, B.K. Anonymization of court orders. In Proceedings of the 2016 11th Iberian Conference on Information Systems and Technologies (CISTI), Gran Canaria, Spain, 15–18 June 2016; pp. 1–4.
10. Tamper, M.; Oksanen, A.; Tuominen, J.; Hyvönen, E.; Hietanen, A. Anonymization Service for Finnish Case Law: Opening Data without Sacrificing Data Protection and Privacy of Citizens. In Proceedings of the International Conference on Law via the Internet, LVI, Florence, Italy, 11–12 October 2018.
11. Vokinger, K.N.; Stekhoven, D.J.; Krauthammer, M. Lost in Anonymization—A Data Anonymization Reference Classification Merging Legal and Technical Considerations. *J. Law Med. Ethics* **2020**, *48*, 228–231. [CrossRef]
12. Pseudonymization according to the GDPR [Definitions and Examples]. Available online: https://dataprivacymanager.net/pseudonymization-according-to-the-gdpr/ (accessed on 15 February 2021).
13. Pseudonymization vs. Anonymization: GDPR. Available online: https://www.tokenex.com/blog/general-data-protection-regulation-pseudonymization-vs-anonymization (accessed on 1 June 2021).
14. Chen, B.; Kifer, D.; LeFevre, K.; Machanavajjhala, A. Privacy-Preserving Data Publishing. *Found. Trends Databases* **2009**, *2*, 1–167. [CrossRef]
15. Sweeney, L. Computational Disclosure Control: A Primer on Data Privacy Protection. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2001.
16. Prasser, F.; Eicher, J.; Spengler, H.; Bild, R.; Kuhn, K.A. Flexible data anonymization using ARX—Current status and challenges ahead. *Softw. Pract. Exp.* **2020**, *50*, 1277–1304. [CrossRef]

17. Tamper, M.; Oksanen, A.; Tuominen, J.; Hietanen, A.; Hyvönen, E. Automatic annotation service appi: Named entity linking in legal domain. In Proceedings of the European Semantic Web Conference, Heraklion, Crete, Greece, 6–10 June 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 208–213.
18. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
19. Vokinger, K.N.; Mühlematter, U.J. Re-Identifikation von Gerichtsurteilen Durch "Linkage" von Daten (banken): Eine Empirische Analyse Anhand von Bundesgerichtsbeschwerden Gegen (Preisfestsetzungs-) Verfügungen von Arzneimitteln. 2019. Available online: https://jusletter.weblaw.ch/juslissues/2019/990/re-identifikation-vo_21cb82c096.html__ONCE&login=false (accessed on 1 June 2021).
20. Narayanan, A.; Shmatikov, V. How to break anonymity of the netflix prize dataset. *arXiv* **2006**, arXiv:cs/0610105.
21. Narayanan, A.; Shmatikov, V. Robust de-anonymization of large sparse datasets. In Proceedings of the 2008 IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA, USA, 18–22 May 2008; pp. 111–125.
22. Sweeney, L. Foundations of privacy protection from a computer science perspective. In Proceedings of the Joint Statistical Meeting, AAAS, Indianapolis, IN, USA, 13–17 August 2000.
23. "Amnesia", A Data Anonymization Tool Supported by the Institute for the Management of Information Systems, 2021. Available online: https://amnesia.openaire.eu/installation.html (accessed on 22 February 2021).
24. Motwani, R.; Nabar, S.U. Anonymizing unstructured data. *arXiv* **2008**, arXiv:0810.5582.
25. Kleinberg, B.; Mozes, M.; van der Toolen, Y. NETANOS-Named Entity-Based Text Anonymization for Open Science. 2017. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjEpbfG553yAhWFNaYKHRI2DHwQFnoECAUQAw&url=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F326121258_NETANOS_-_Named_entity-based_Text_Anonymization_for_Open_Science&usg=AOvVaw0Rlkb0yu7TugFx-LTS6cY9 (accessed on 1 June 2021).
26. Kleinberg, B.; Mozes, M. Web-based text anonymization with Node. js: Introducing NETANOS (Named entity-based Text Anonymization for Open Science). *J. Open Source Softw.* **2017**, *2*, 293. [CrossRef]
27. Mozes, M.; Kleinberg, B. No Intruder, no Validity: Evaluation Criteria for Privacy-Preserving Text Anonymization. *arXiv* **2021**, arXiv:2103.09263.
28. European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data and Repealing Directive 95/46/EC. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwinkevo553yAhVJBKYKHQ5-AGoQFnoECAUQAw&url=https%3A%2F%2Feur-lex.europa.eu%2Feli%2Freg%2F2016%2F679%2Foj&usg=AOvVaw1XAG3mHMtSjUcR1oFXnGgW (accessed on 1 June 2021).
29. Montana Knowledge Management, Ltd. LEXPERT Database of Hungarian Court Decisions. Available online: https://lexpert.hu/ (accessed on 7 August 2021).
30. Dwork, C.; Smith, A.; Steinke, T.; Ullman, J. Exposed! a survey of attacks on private data. *Annu. Rev. Stat. Its Appl.* **2017**, *4*, 61–84. [CrossRef]
31. Microsoft. Differential Privacy for Everyone. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwix8bWi6J3yAhVlxosBHSRxAgMQFnoECAMQAw&url=https%3A%2F%2Fdownload.microsoft.com%2Fdownload%2FD%2F1%2FF%2FD1F0DFF5-8BA9-4BDF-8924-7816932F6825%2FDifferential_Privacy_for_Everyone.pdf&usg=AOvVaw11fKVVmW3XHZZjqLEnXgeR (accessed on 4 February 2021).
32. Zhu, T.; Li, G.; Zhou, W.; Philip, S.Y. *Differential Privacy and Applications*; Springer: Berlin/Heidelberg, Germany, 2017.
33. Narayanan, A.; Shmatikov, V. Robust de-anonymization of large sparse datasets: A decade later. *May* **2019**, *21*, 2019. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwitiZDK6J3yAhVLL6YKHeIxBc0QFnoECA8QAw&url=https%3A%2F%2Fwww.semanticscholar.org%2Fpaper%2FRobust-de-anonymization-of-large-sparse-datasets-%253A-Narayanan-Shmatikov%2Ff41ef0fe589fdfbfe22c1ac5629638773f8d9fe9&usg=AOvVaw0hEX8iw1U22ZfIY-ZZhy5n (accessed on 1 June 2021).
34. Datta, A.; Sharma, D.; Sinha, A. Provable de-anonymization of large datasets with sparse dimensions. In Proceedings of the International Conference on Principles of Security and Trust, Tallinn, Estonia, 24 March–1 April 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 229–248.
35. Dalenius, T. Finding a needle in a haystack or identifying anonymous census records. *J. Off. Stat.* **1986**, *2*, 329.
36. El Emam, K.; Jabbouri, S.; Sams, S.; Drouet, Y.; Power, M. Evaluating common de-identification heuristics for personal health information. *J. Med Internet Res.* **2006**, *8*, e28. [CrossRef] [PubMed]
37. El Emam, K.; Jonker, E.; Sams, S.; Neri, E.; Neisa, A.; Gao, T.; Chowdhury, S. *Pan-Canadian De-Identification Guidelines for Personal Health Information*; Privacy Commissioner of Canada: Gatineau, QC, Canada, 2007.
38. El Emam, K.; Brown, A.; AbdelMalik, P. Evaluating predictors of geographic area population size cut-offs to manage re-identification risk. *J. Am. Med Inform. Assoc.* **2009**, *16*, 256–266. [CrossRef]
39. Canadian Institutes of Health Research Privacy Advisory Committee. *CIHR Best Practices for Protecting Privacy in Health Research, September 2005*; PublicWorks and Government Services Canada: Ottawa, ON, Canada, 2005.
40. Sweeney, L. Simple demographics often identify people uniquely. *Health* **2000**, *671*, 1–34.
41. Hafner, K. If you liked the movie, a Netflix contest may reward you handsomely. *N. Y. Times* **2006**, 2.

42. Barbaro, M.; Zeller, T.; Hansell, S. A face is exposed for AOL searcher no. 4417749. *N. Y. Times* **2006**, *9*, 8.
43. Zhu, T.; Li, G.; Zhou, W.; Philip, S.Y. Preliminary of differential privacy. In *Differential Privacy and Applications*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 7–16.
44. Dankar, F.K.; El Emam, K.; Neisa, A.; Roffey, T. Estimating the re-identification risk of clinical data sets. *BMC Med Inform. Decis. Mak.* **2012**, *12*, 1–15. [CrossRef]
45. Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; Naor, M. Our data, ourselves: Privacy via distributed noise generation. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, 8–12 May 2016; Springer: Berlin/Heidelberg, Germany, 2006; pp. 486–503.
46. Prasser, F.; Kohlmayer, F.; Kuhn, K.A. The importance of context: Risk-based de-identification of biomedical data. *Methods Inf. Med.* **2016**, *55*, 347–355.
47. Domingo-Ferrer, J.; Torra, V. A critique of k-anonymity and some of its enhancements. In Proceedings of the 2008 Third International Conference on Availability, Reliability and Security, Barcelona, Spain, 4–7 March 2008; pp. 990–993.
48. Nergiz, M.E.; Atzori, M.; Clifton, C. Hiding the presence of individuals from shared databases. In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, Beijing, China, 11–14 June 2007; pp. 665–676.
49. El Emam, K.; Arbuckle, L. *Anonymizing Health Data: Case Studies and Methods to Get You Started*; O'Reilly Media, Inc.: Newton, MA, USA, 2013.
50. Truta, T.M.; Vinay, B. Privacy protection: P-sensitive k-anonymity property. In Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06), Atlanta, GA, USA, 3–7 April 2006; p. 94.
51. Samarati, P.; Sweeney, L. Protecting Privacy when Disclosing Information: K-Anonymity and Its Enforcement through Generalization and Suppression. 1998. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjeos6j6Z3yAhXDDaYKHTTnABMQFnoECAQQAw&url=https%3A%2F%2Fepic.org%2Fprivacy%2Freidentification%2FSamarati_Sweeney_paper.pdf&usg=AOvVaw0UDf7utmmgKAgkKhXNegKB (accessed on 1 June 2021).
52. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. l-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data (TKDD)* **2007**, *1*, 3–es. [CrossRef]
53. Li, N.; Li, T.; Venkatasubramanian, S. t-closeness: Privacy beyond k-anonymity and l-diversity. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2007; pp. 106–115.
54. Samarati, P. Protecting respondents identities in microdata release. *IEEE Trans. Knowl. Data Eng.* **2001**, *13*, 1010–1027. [CrossRef]
55. Sweeney, L. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **2002**, *10*, 557–570. [CrossRef]
56. Sweeney, L. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **2002**, *10*, 571–588. [CrossRef]
57. Rubner, Y.; Tomasi, C.; Guibas, L.J. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vis.* **2000**, *40*, 99–121. [CrossRef]
58. Vico, H.; Calegari, D. Software architecture for document anonymization. *Electron. Notes Theor. Comput. Sci.* **2015**, *314*, 83–100. [CrossRef]
59. UTD Anonymization Toolbox. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjrxazI6Z3yAhWawosBHZR8CyIQFnoECAUQAw&url=http%3A%2F%2Fwww.cs.utdallas.edu%2Fdspl%2Ftoolbox%2F&usg=AOvVaw1-orTi0L4d9eKuAGLNRtEA (accessed on 1 June 2021).
60. Sánchez, D.; Martínez, S.; Domingo-Ferrer, J.; Soria-Comas, J.; Batet, M. μ-ANT: Semantic microaggregation-based anonymization tool. *Bioinformatics* **2020**, *36*, 1652–1653. [CrossRef] [PubMed]
61. Cornell Anonymization Toolkit. Available online: https://sourceforge.net/projects/anony-toolkit/ (accessed on 15 February 2021).
62. Dai, C.; Ghinita, G.; Bertino, E.; Byun, J.W.; Li, N. TIAMAT: A tool for interactive analysis of microdata anonymization techniques. *Proc. VLDB Endow.* **2009**, *2*, 1618–1621. [CrossRef]
63. Poulis, G.; Gkoulalas-Divanis, A.; Loukides, G.; Skiadopoulos, S.; Tryfonopoulos, C. Secreta: A tool for anonymizing relational, transaction and rt-datasets. In *Medical Data Privacy Handbook*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 83–109.
64. Prasser, F.; Kohlmayer, F. Putting statistical disclosure control into practice: The ARX data anonymization tool. In *Medical Data Privacy Handbook*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 111–148.
65. Bild, R.; Kuhn, K.A.; Prasser, F. Better Safe than Sorry–Implementing Reliable Health Data Anonymization. *Stud. Health Technol. Inform.* **2020**, *270*, 68–72.
66. Gardner, J.; Xiong, L. HIDE: An integrated system for health information DE-identification. In Proceedings of the 2008 21st IEEE International Symposium on Computer-Based Medical Systems, Jyvaskyla, Finland, 17–19 June 2008; pp. 254–259.
67. Gardner, J.; Xiong, L. An integrated framework for de-identifying unstructured medical data. *Data Knowl. Eng.* **2009**, *68*, 1441–1451. [CrossRef]
68. Kifer, D.; Machanavajjhala, A. No free lunch in data privacy. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, Athens, Greece, 12–16 June 2011; pp. 193–204.

69. Wolpert, D.H. The supervised learning no-free-lunch theorems. *Soft Comput. Ind.* **2002**, 25–42. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi8j4_q6Z3yAhWIOpQKHU4bCq8QFnoECAQQAw&url=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F229078412_The_Supervised_Learning_No-Free-Lunch_Theorems&usg=AOvVaw0EcJCII1hyqvybrgKDOtqy (accessed on 1 June 2021).
70. Gómez, D.; Rojas, A. An empirical overview of the no free lunch theorem and its effect on real-world machine learning classification. *Neural Comput.* **2016**, *28*, 216–228. [CrossRef] [PubMed]
71. El Emam, K. *Guide to the De-Identification of Personal Health Information*; CRC Press: Boca Raton, FL, USA, 2013.
72. El Emam, K. Risk-based de-identification of health data. *IEEE Secur. Priv.* **2010**, *8*, 64–67. [CrossRef]
73. Newcombe, H.B.; Kennedy, J.M.; Axford, S.; James, A.P. Automatic linkage of vital records. *Science* **1959**, *130*, 954–959. [CrossRef] [PubMed]
74. Fellegi, I.P.; Sunter, A.B. A theory for record linkage. *J. Am. Stat. Assoc.* **1969**, *64*, 1183–1210. [CrossRef]
75. Schlörer, J. Identification and retrieval of personal records from a statistical data bank. *Methods Inf. Med.* **1975**, *14*, 7–13. [CrossRef]
76. Schilder, F. Event extraction and temporal reasoning in legal documents. In *Annotating, Extracting and Reasoning about Time and Events*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 59–71.
77. Lagos, N.; Segond, F.; Castellani, S.; O'Neill, J. Event extraction for legal case building and reasoning. In Proceedings of the International Conference on Intelligent Information Processing, Manchester, UK, 13–16 October 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 92–101.
78. GDPR Implementation of Denmark, 2020. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjysPmK6p3yAhUDy4sBHZ8EBlUQFnoECAUQAw&url=https%3A%2F%2Fwww.opengovpartnership.org%2Fdocuments%2Fdenmark-implementation-report-2017-2019%2F&usg=AOvVaw2T8SMjFedIersA1V4Xa5pd (accessed on 1 June 2021).
79. OpenAIRE Webinar—Amnesia, an Open-Source, Flexible Data Anonymization Tool, 2020. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwitsu6o6p3yAhVMIqYKHWpbCSAQFnoECAkQAw&url=https%3A%2F%2Famnesia.openaire.eu%2F&usg=AOvVaw0yXvLpjIfh7g5MrhDLXNN_ (accessed on 1 June 2021).
80. Központi Statisztikai Hivatal (Central Office of Statistics). *Population Data in Békés Country by Sexes and Age Groups*; Central Office of Statistics: Budapest, Hungary, 2020.
81. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
82. Joyce, J. Bayes' Theorem. 2003. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjDyufr6p3yAhXRw4sBHd6WCboQFnoECAQQAw&url=https%3A%2F%2Fccc.inaoep.mx%2F~villasen%2Findex_archivos%2FcursoTATII%2FEntidadesNombradas%2FSekine-%2520NEsHistory04.pdf&usg=AOvVaw1Rr_qUVeVK2_ycH15cxTbd (accessed on 1 June 2021).
83. MacKay, D.J.; Mac Kay, D.J. *Information Theory, Inference and Learning Algorithms*; Cambridge University Press: Cambridge, UK, 2003; pp. 23–24, 138–139.
84. Chinchor, N.; Robinson, P. MUC-7 named entity task definition. In Proceedings of the 7th Conference on Message Understanding, Fairfax, VA, USA, 29 April–1 May 1997; Volume 29, pp. 1–21.
85. Sekine, S. Named Entity: History and Future. 2004. Available online: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiok7XW653yAhWBHKYKHaVRAAAQFnoECAQQAw&url=https%3A%2F%2Fccc.inaoep.mx%2F~villasen%2Findex_archivos%2FcursoTATII%2FEntidadesNombradas%2FSekine-%2520NEsHistory04.pdf&usg=AOvVaw1Rr_qUVeVK2_ycH15cxTbd (accessed on 1 June 2021).
86. Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *Lingvisticae Investig.* **2007**, *30*, 3–26. [CrossRef]
87. Szarvas, G.; Farkas, R.; Kocsor, A. A multilingual named entity recognition system using boosting and c4. 5 decision tree learning algorithms. In Proceedings of the International Conference on Discovery Science, Barcelona, Spain, 7–10 October 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 267–278.
88. Dernoncourt, F.; Lee, J.Y.; Uzuner, O.; Szolovits, P. De-identification of patient notes with recurrent neural networks. *J. Am. Med Inform. Assoc.* **2017**, *24*, 596–606. [CrossRef]
89. Iglesias, A.; Castro, E.; Pérez, R.; Castaño, L.; Martínez, P.; Gómez-Pérez, J.M.; Kohler, S.; Melero, R. Mostas: Un etiquetador morfo-semántico, anonimizador y corrector de historiales clínicos. *Proces. Del Leng. Nat.* **2008**, *41*.
90. Bagga, A.; Baldwin, B. Entity-Based Cross-Document Core f erencing Using the Vector Space Model. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Montréal, QC, Canada, 10–14 August 1998; pp. 79–85.
91. Boros, E. Neural Methods for Event Extraction. Ph.D. Thesis, Université Paris-Saclay, Yvette, France, 2018.
92. Shen, S.; Qi, G.; Li, Z.; Bi, S.; Wang, L. Hierarchical Chinese Legal event extraction via Pedal Attention Mechanism. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 13–18 September 2020; pp. 100–113.
93. Subecz, Z. Event detection and classification in hungarian natural texts. *Eur. Sci. J.* **2019**, *15*. [CrossRef]
94. He, L.; Lee, K.; Lewis, M.; Zettlemoyer, L. Deep semantic role labeling: What works and what is next. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 473–483.
95. Office, I.C. Anonymisation: Managing data protection risk code of practice. *ICO* **2012**.

symmetry

MDPI

Article

Fast Computation of Green Function for Layered Seismic Field via Discrete Complex Image Method and Double Exponential Rules

Siqin Liu [1,2], Zhusheng Zhou [1,2,*], Shikun Dai [1,2], Ibrar Iqbal [3] and Yang Yang [3]

[1] School of Geosciences and Info-Physics, Central South University, Changsha 410083, China; liusiqin@csu.edu.cn (S.L.); dskgmes@csu.edu.cn (S.D.)
[2] Key Laboratory of Metallogenic Prediction of Nonferrous Metals and Geological Environment Monitoring, Ministry of Education, Changsha 410083, China
[3] College of Earth Sciences, Guilin University of Technology, Guilin 541004, China; ibrariqbal@glut.edu.cn (I.I.); yangyang@glut.edu.cn (Y.Y.)
* Correspondence: geophys@126.com or 175001022@csu.edu.cn

Abstract: A novel computational method to evaluate the Sommerfeld integral (SI) efficiently and accurately is presented. The method rewrites the SI into two parts, applying discrete complex image method (DCIM) to evaluate the infinite integral while using double exponential quadrature rules (DE rules) for the computation of the finite part. Estimation of signal parameters via rotational invariance techniques (ESPRIT) is used to improve the accuracy and efficiency of extracting DCIM compared to the generalized pencil of function (GPOF). Due to the symmetry of the horizontal layered media, the Green function, representing the seismic fields due to a point source, can be written in the form of Sommerfeld integral in cylindrical coordinate system and be calculated by the proposed method. The performance of the method is then compared to the DE rules with weighted average partition extrapolation (WA), which shows a good agreement, with computational time reduced by about 40%.

Keywords: DE rules; Green function; DCIM; Sommerfeld integral

Citation: Liu, S.; Zhou, Z.; Dai, S.; Iqbal, I.; Yang, Y. Fast Computation of Green Function for Layered Seismic Field via Discrete Complex Image Method and Double Exponential Rules. *Symmetry* **2021**, *13*, 1969. https://doi.org/10.3390/sym13101969

Academic Editors: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Received: 23 September 2021
Accepted: 17 October 2021
Published: 19 October 2021

1. Introduction

Green function for the horizontal layered seismic field is usually derived by reflectivity method, which was proposed by Fuchs and Müller [1] and extended to many other kinds [2], like reflection and transmission coefficient matrix method [3], discrete wavenumber method [4], discrete wavenumber finite element method [5], and generalized reflection transmission coefficient matrix method [6]. In the frequency domain, the Green function, derived by the reflectivity method, can be written in the Sommerfeld integral form in cylindrical coordinate system for symmetrical media.

It is well known that the numerical evaluation of Sommerfeld integral (SI) is computationally expensive due to the oscillatory and slow convergence of the integrands. To overcome this problem, several approaches have been proposed, which can be divided into two main categories: one is the approximation of the spatial domain Green functions in a closed form where no numerical integration is needed, and the other is the numerical integration of SI in conjunction with some acceleration techniques [7]. Within the first category, discrete complex image method (DCIM), which approximates the integrand of Sommerfeld integral by a series of complex exponential functions, is commonly used for the advantages of high computational efficiency, but it needs to handle the surface wave poles contributions, which not only makes the computation complicated but also brings singularity to the near region [8], and also the calculation accuracy and effective range are difficult to be accurately estimated. For the latter category, the common practice is dividing the whole Sommerfeld integral into two parts: the first part is the path to bypass the singularity; the second part, the path to infinity, is the Sommerfeld tail integral. The finite-range integrals may readily be evaluated by the Gauss–Jacobi quadrature [9] or by the double-exponential (DE) rule [10,11]. Mosig first used DE rules to calculate Sommerfeld integrals in [11,12] which indicated its validity of suppressing endpoint singularities. The calculation of tail integral is difficult to converge due to Bessel's oscillation and slow attenuation characteristics, so this kind of method generally requires extrapolation to accelerate convergence. The WA method has shown higher levels of convergence among various extrapolation methods [13–15]. This kind of method does not need to strictly locate the position of singularity in Sommerfeld integral but only needs to ensure that the first integral path avoids all the singularities. It has good adaptability and controllable numerical accuracy; however, this depends on the number of intervals (n) that are chosen to evaluate the tail region. The computational time also rapidly increases as the value of (n) increases [16]. Given the advantages and disadvantages of these two methods, we propose a method by combining DE rules and DCIM to calculate Sommerfeld integral.

This paper first presents the Green function of a point source in a multilayer half-space in Section 3, explaining the mathematical manipulation required to obtain the solution as a Sommerfeld integral form in the frequency domain; it then describes the principle of the proposed method with DE quadrature rules and DCIM in Section 4; finally, it corroborates the correctness of the algorithm by the frequency responses obtained from the proposed approach with those where DE rules and WA partition-extrapolation are used for half-space model, and the finite element method is used for three layers model.

2. Seismic Wave Equation and Green Function

2.1. Seismic Wave Equation

The propagation of seismic wavefield in the time domain can be simplified by the following three-dimensional acoustic wave equation:

$$\nabla^2 u(x,y,z,t) = \frac{1}{v(x,y,z)^2}\frac{\partial^2 u(x,y,z,t)}{\partial t^2} + f(x,y,z,t) \tag{1}$$

where $u\ (x,y,z,t)$, $v\ (x,y,z,t)$, and $f(x,y,z,t)$ represent displacement, velocity, and source term, respectively. $f(x,y,z,t) = -\delta(x-x_s, y-y_s, z-z_s)s(t)$, $s(t)$ is the wavelet, and Ricker wavelet is used in this paper; and $\delta(x-x_s, y-y_s, z-z_s)$ is the Dirac function at the source point (x_s, y_s, z_s).

By Fourier-transform of Equation (1), the two-dimensional acoustic wave equation in frequency domain is obtained, and therefore, Green function for the problem is defined by the following equation:

$$\nabla^2 G(x,y,z,\omega) + k^2 G(x,y,z,\omega) = F(x,y,z,\omega) \tag{2}$$

where, G denotes the Green function, $F(x,y,z,\omega) = -\delta(x-x_s, y-y_s, z-z_s)S(\omega)$ is the source term in the frequency domain, $k(x,y,z) = \omega/v$ is wave number, $S(\omega)$ is Ricker wavelet in the frequency domain, and ω is the angular frequency.

However, the underground medium is always viscous, which leads to wave energy loss and phase change in the process of propagation. The visco-acoustic wave equation is established to better describe the propagation of the seismic waves in this viscous medium, which is the same form as Equation (2), but the complex velocity is introduced to simulate the viscous effect [17–19]. The reciprocal of complex velocity is defined as $\frac{1}{\tilde{v}} = \frac{1}{v}\left(1 - \frac{\mathrm{j}}{2Q}\right)$ [18], where Q is quality factor, so the complex wavenumber is set to be $k = \frac{\omega}{v}\left(1 - \frac{\mathrm{j}}{2Q}\right), \mathrm{j} = \sqrt{-1}$. In this paper, the value Q generated by Li Qingzhong's empirical formula is used for numerical simulation [20]

$$Q = 14 \times (v/1000.0)^{2.2} \tag{3}$$

2.2. Green Function in Full-Space

The Green function of Equation (2) in homogeneous full-space can be expressed as

$$G(x,y,z,\omega) = \frac{S(\omega)e^{-\mathrm{i}k_1R}}{4\pi R} \tag{4}$$

where $R = \sqrt{(x-x_s)^2 + (y-y_s)^2 + (z-z_s)^2}$, $S(\omega)$ is Ricker wavelet in the frequency domain, ω is the angular frequency, and k_1 is the wavenumber of the medium. The above formula can be rewritten in the Sommerfeld integral form in cylindrical coordinate system:

$$\frac{S(\omega)e^{-\mathrm{i}k_1R}}{4\pi R} = \frac{S(\omega)}{4\pi}\int_0^\infty \frac{m}{m_1}e^{-m_1|z-z_s|}J_0(mr)dm \tag{5}$$

where $r = \sqrt{(x-x_s)^2 + (y-y_s)^2}$, $m_1 = \sqrt{m^2 - k_1{}^2}$.

2.3. Green Function in Layered Half-Space

Consider n layers symmetric structure of homogeneous medium defined by interfaces located at $z_1, z_2, \cdots, z_{n-1}$, as shown in Figure 1. The density of each layer is $\rho_1, \rho_2, \cdots, \rho_n$; and the velocity is $v_1, v_2, \cdots, v_n$. In this paper, the source is placed in the second layer.

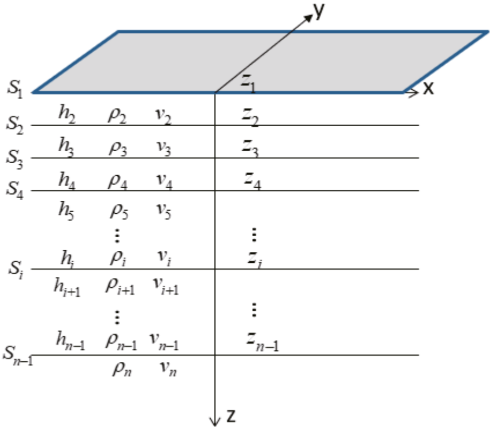


Figure 1. *n* layers structure of homogeneous medium.

Each layer satisfies the acoustic Equation (2) with parameters of the Green function G, wavenumber k, velocity v, density ρ, layer thickness h, and quality factor Q, respectively. We obtain the equations as follows:

$$\nabla^2 G_i + k_i{}^2 G_i = 0, \quad i = 1,3,\cdots,n \tag{6}$$

$$\nabla^2 G_i + k_i{}^2 G_i = -S(\omega)\delta(R - R_0) \quad i = 2 \tag{7}$$

where $k_i = \frac{\omega}{v_i}\left(1 - \frac{\mathrm{j}}{2Q}\right)$, $\mathrm{j} = \sqrt{-1}$.

At the interface, the pressure $P_i = \rho_i G_i$ as well as the gradient of the potential for the vertical direction $\frac{\partial G_i}{\partial z_i}$ are continuous [21]. Therefore, the following boundary conditions can be imposed on the Green function

$$\frac{\partial G_i}{\partial z} = \frac{\partial G_{i+1}}{\partial z}, \rho_i G_i = \rho_{i+1}G_{i+1}, (i = 1,2,\cdots,n-1) \tag{8}$$

The solutions of Equations (6) and (7) can be regarded as the summation of separate up-going and down-going waves, and therefore, it can be written in the form of Sommerfeld integral in cylindrical coordinate system as follows:

$$G_i = \frac{S(\omega)}{4\pi}\int_0^{\infty}\left(C_i e^{m_i z} + D_i e^{-m_i z}\right)J_0(mr)dm, \quad i = 1, 3, \cdots, n \tag{9}$$

$$G_i = \frac{S(\omega)}{4\pi}\left[\frac{e^{-\mathrm{i}k_i R}}{R} + \int_0^{\infty}\left(C_i e^{m_i z} + D_i e^{-m_i z}\right)J_0(mr)dm\right], \quad i = 2 \tag{10}$$

where $m_i = \sqrt{m^2 - k_i^2}$, $k_i^2 = \frac{\omega^2}{\widetilde{v}_i{}^2}$, $\widetilde{v}_i = v_i\left(1 - \frac{\mathrm{j}}{2Q_n}\right)$, $i = 1, 2, \cdots, n$, $\mathrm{j} = \sqrt{-1}$. In Equations (9) and (10), $e^{m_i z}$ and $e^{-m_i z}$ may be infinity. To maintain numerical stability, rewrite Equations (9) and (10) into

$$G_i = \frac{S(\omega)}{4\pi}\int_0^{\infty}\left(C_i e^{m_i(z-z_i)} + D_i e^{-m_i(z-z_{i-1})}\right)J_0(mr)dm, \quad i = 1, 3, \cdots, n \tag{11}$$

$$G_i = \frac{S(\omega)}{4\pi}\left[\frac{e^{-\mathrm{i}k_i R}}{R} + \int_0^{\infty}\left(C_i e^{m_i(z-z_i)} + D_i e^{-m_i(z-z_{i-1})}\right)J_0(mr)dm\right], \quad i = 2 \tag{12}$$

By using the boundary conditions (8), the unknowns $C_1, C_2, D_2, \cdots, C_i, D_i, \cdots, C_{n-1}, D_{n-1}, D_n$ in the above formula are solved. The coefficients of the source layer are derived firstly, and other coefficients can be obtained by recursion; then, the expression of the Green function of the layered medium is obtained.

$$D_2 = \frac{m}{m_2}\frac{H_2^{d*}\left(H_3^{u*}e^{-m_2|z_2-z_s|}e^{-m_2 h_2} - e^{-m_2|z_1-z_s|}\right)}{1 - H_3^{u*}H_2^{d*}e^{-2m_2 h_2}} \tag{13}$$

$$C_2 = -H_3^{u*}\left(\frac{m}{m_2}e^{-m_2|z_2-z_s|} + D_2 e^{-m_2 h_2}\right) \tag{14}$$

$$D_i = \frac{D_{i-1}m_{i-1}e^{-m_{i-1}h_{i-1}}\left(1 + H_i^{u*}\right)}{m_i\left(1 + H_{i+1}^{u*}e^{-2m_i h_i}\right)} \tag{15}$$

$$C_i = -H_{i+1}^{u*}D_i e^{-m_i h_i} \tag{16}$$

where $h_i = z_{i+1} - z_i$, $H_2^d = \frac{\rho_1 m_2}{\rho_2 m_1}$, $H_2^{d*} = \frac{1-H_2^d}{1+H_2^d}$, $H_n^u = \frac{\rho_n m_{n-1}}{\rho_{n-1}m_n}$, $H_n^{u*} = \frac{(1-H_n^u)}{(1+H_n^u)}$, $H_{i+2}^{u*} = \frac{\left(1-H_{i+2}^u\right)}{\left(1+H_{i+2}^u\right)}$, $H_{i+2}^u = \frac{\rho_{i+2}m_{i+1}}{m_{i+2}\rho_{i+1}}\frac{\left(1-H_{i+3}^{u*}e^{-2m_{i+2}h_{i+2}}\right)}{\left(1+H_{i+3}^{u*}e^{-2m_{i+2}h_{i+2}}\right)}$, $i = 3, \cdots, n-1$.

3. Methods

The expressions (11) and (12) contain Sommerfeld integrals, which is an infinite integral with the highly oscillatory and slow-decaying kernel. In this paper, the partial closed form of Sommerfeld integral is derived, and ESPRT is applied to extract DCIM, while DE rules are used for the computation of the finite integration.

3.1. Partial Closed Form Expression

The Sommerfeld integration in Equations (11) and (12) can be written in the following form:

$$I = \int_0^{\infty}\frac{m}{m_i}g(m_i)e^{-m_i|z-z_s|}J_0(mr)dm = \int_0^{\infty}\Gamma_m J_0(mr)dm \tag{17}$$

The integration is then divided into two parts,

$$\int_0^\infty \Gamma_m J_0(mr)dm = \int_0^p \Gamma_m J_0(mr)dm + \int_p^\infty \Gamma_m J_0(mr)dm \tag{18}$$

where p is a reasonably selected integral breakpoint. The second integral on the right side of the Equation (18) can be asymptotically approximated and then be written as

$$\int_p^\infty \Gamma_m J_0(mr)dm \approx \int_0^\infty \Gamma_m^\infty J_0(mr)dm - \int_0^p \Gamma_m^\infty J_0(mr)dm \tag{19}$$

On substituting (19) in (18), we get

$$I \approx \int_0^p (\Gamma_m - \Gamma_m^\infty) J_0(mr)dm + \int_0^\infty \Gamma_m^\infty J_0(mr)dm \tag{20}$$

The kernel function $g(m_i)$ in (17) can be approximated by an exponential function,

$$\Gamma_m^\infty = g(m_i) \cdot \frac{m}{m_i} e^{-m_i|z-z'|} = \sum_{l=1}^{pb} a(l) \exp[b(l)m_i] \cdot \frac{m}{m_i} e^{-m_i|z-z'|} \tag{21}$$

where pb is the number of exponentials used for approximation. In this paper, the coefficients $a\ (l)$, $b\ (l)$ are solved by the ESPRIT algorithm, which will be discussed in the next section.

From the Sommerfeld identity expressed by formula (5), the closed form of the second integral can be obtained,

$$\int_0^\infty \Gamma_m^\infty J_0(mr)dm = \sum_{l=1}^{pb} a\ (l) \frac{\exp(-\mathrm{j}k_i R_l)}{R_l} \tag{22}$$

where $R_l = \sqrt{r^2 + [b\ (l) - |z - z_s|]^2}$, $r = \sqrt{(x - x_s)^2 + (y - y_s)^2}$, (x_s, y_s, z_s) is the location of the source.

The first part of the $g(m_i)$ function usually contains singularity, and the tail is smooth and decays fast. Therefore, if the appropriate p value is selected, the approximate fitting will be very accurate by avoiding the singular value in the front part. The finite integral with singularity in (20) can be evaluated directly by the numerical integration method. We choose DE quadrature rules here for integration with the advantage of dealing with singular points and high precision.

3.2. ESPRIT Algorithm

The signal $g(m_i)$ can be sampled as [8]

$$m_i = k_i \left[T_{02} + \frac{T_{01} - T_{02}}{T_{01}} t \right], 0 \le t \le T_{01} \tag{23}$$

The value of T_{01} can be selected from 100~200, T_{02} usually set between 1~3, and in this paper, $T_{01} = 200$, $T_{02} = 2$, t is an integer. According to the relationship $m^2 = m_i^2 + k_i^2$, the first sampling in m-plane is $k_i\sqrt{1 + T_{02}^2}$. Therefore, the approximation of $g(m_i)$ starts from $m = k_i\sqrt{1 + T_{02}^2}$, and the parameter p in formula (20) should be set not less than $k_i\sqrt{1 + T_{02}^2}$ to insure the integration accuracy.

Then the sampling sequence can be expressed as

$$g(m_i) = y(t) \approx \sum_{l=1}^{pb} A\ (l) \exp[B\ (l)t] \tag{24}$$

The relation of $A(l)$, $B(l)$, and $a\ (l)$, $b\ (l)$ can be obtained from Equations (23) and (24).

$$\sum_{l=1}^{pb} a\ (l)\exp[b\ (l)m_i] = \sum_{l=1}^{pb} A(l)\exp[B(l)t] \tag{25}$$

Then, the unknown coefficients $a\ (l)$, $b\ (l)$ in (24) are obtained.

$$b\ (l) = \frac{B\ (l)T_{01}}{k_i(T_{01} - T_{02})} \tag{26}$$

$$a\ (l) = A\ (l)\exp(b\ (l)\cdot k_i \cdot T_{02}) \tag{27}$$

For the sampling sequence $y(0), y(1), \cdots, y(N-1)$, a data matrix can be constructed:

$$\mathbf{Y} = \begin{bmatrix} y(0) & y(1) & \cdots & y(L) \\ y(1) & y(2) & \cdots & y(L+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(N-L-1) & y(N-L) & \cdots & y(N-1) \end{bmatrix}_{(N-L)\times(L+1)} \tag{28}$$

where N is the sample number, L is called the pencil parameter, and its value should be between $N/3$ and $N/2$ [22].The data matrix $\mathbf{Y}$ can be decomposed by SVD,

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H = \begin{bmatrix} \mathbf{U}_s & \mathbf{U}_n \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_s & 0 \\ 0 & \mathbf{\Sigma}_n \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^H \\ \mathbf{V}_n^H \end{bmatrix} \tag{29}$$

where $\mathbf{U}$ is $(N-L)\times(N-L)$ orthogonal matrix, and $\mathbf{V}$ is $(L+1)\times(L+1)$ orthogonal matrix, $\mathbf{\Sigma}$ is $(N-L)\times(L+1)$ diagonal matrix with main diagonal element σ_l, which is the singular value of matrix $\mathbf{Y}$. For signals without noise, $\mathbf{Y}$ has pb non-zero singular values σ_l (l= 1,2,..., pb), and pb represents the highest order of the exponential signal of formula (24). If the signal contains noise, mode number pb can be recorded by setting a minimum threshold for σ_l.

Take out the first pb dominant right singular vectors in $\mathbf{V}_s$ matrix to form $(L+1)\times pb$ matrix $\mathbf{V}_s^{pb}$.The last line of $\mathbf{V}_s^{pb}$ is deleted to obtain $L \times M$ matrix $\mathbf{V}_1$; the first line of $\mathbf{V}_s^{pb}$ is deleted to get $L \times M$ matrix $\mathbf{V}_2$. Construct a matrix $\mathbf{\Psi}$ [23]

$$\mathbf{\Psi} = \left(\mathbf{V}_1^H\mathbf{V}_1\right)^{-1}\mathbf{V}_1^H\mathbf{V}_2 \tag{30}$$

Find the eigenvalues λ_l of the matrix $\mathbf{\Psi}$,

$$B\ (l) = \log(\lambda\ _l),\ l = 1, 2, \cdots, pb \tag{31}$$

For N sampled signals,

$$\boldsymbol{\lambda} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_{pb} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_1^{N-1} & \lambda_2^{N-1} & \cdots & \lambda_{pb}^{N-1} \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{pmatrix}, \mathbf{A} = \begin{bmatrix} A(1) & A(2) & \cdots & A(pb) \end{bmatrix} \tag{32}$$

and

$$\mathbf{Y} = \boldsymbol{\lambda}\mathbf{A} \tag{33}$$

According to the least square method, we can obtain

$$\mathbf{A} = \left(\boldsymbol{\lambda}^H\boldsymbol{\lambda}\right)^{-1}\boldsymbol{\lambda}^H\mathbf{Y} \tag{34}$$

So far, the coefficients $A\ (l)$ and $B\ (l)$ are obtained. According to (22), (26), and (27), the second part of Equation (20) is solved, and the integrand of the first part is gained, and then DE rules are applied to compute the finite integral.

3.3. DE Rules

The double exponential transformation was first proposed by Takahasi and Mori in 1974 [16]. It can be seen from Equations (11)–(16) that the integration kernel is singular at $m_i = k_i$. DE rules are insensitive to endpoint singularity and simple to program since the weights and nodes are easily generated [12].

Consider the following form of integral:

$$I = \int_{-1}^{1} ff(\xi)d\xi \tag{35}$$

Let a variable transform:

$$\xi = \varphi(t) \text{ and } \varphi(-\infty) = -1, \varphi(+\infty) = 1 \tag{36}$$

be applied into (35) so as to change the interval $[-1, 1]$ into the infinite interval $[-\infty, +\infty]$

$$I = \int_{-\infty}^{+\infty} ff(\varphi(t))\varphi'(t)dt \tag{37}$$

The DE rules are transformed by the tanh–sinh formula,

$$\varphi(t) = \tanh(gs(t)) = \tanh(\sinh(t)) \tag{38}$$

$$\varphi'(t) = gs'(t)\text{sech}^2 gs(t) = \frac{\cosh(t)}{\cosh^2(\sinh(t))} \tag{39}$$

The standard trapezoidal rule for numerical integration is applied with h as grid interval when the integral is defined on the interval $[-\infty, +\infty]$, and n is the sample point, which is truncated at $\pm N$. Then we can approximate the definite integral via

$$I = h \sum_{n=-\infty}^{\infty} ff(\varphi(nh))\varphi'(nh) \approx h \sum_{n=-N}^{N} \omega_n ff(\xi_n) \tag{40}$$

with the nodes ξ_k and weights ω_k defined as

$$\xi_n = 1 - \delta_n, \omega_n = 2g'(nh)\delta_n(1 - q_n)^{-1} \tag{41}$$

where

$$\delta_n = 2q_n(1 + q_n)^{-1}, q_n = e^{-2gs(nh)} \tag{42}$$

For arbitrary integral interval $[a, b]$ may be mapped onto $[-1, 1]$ by the linear transformation $\xi = \sigma x + \gamma$ with $\sigma = (a - b)/2$, $\gamma = (a + b)/2$, which leads to

$$\int_a^b ff(\xi)d\xi = \sigma \int_{-1}^{1} ff(\sigma x + \gamma)dx \tag{43}$$

Hence, for an arbitrary interval $[a, b]$, the nodes and weights become $\sigma\xi_k + \gamma$ and $\sigma\omega_k$, respectively, and (43) is transformed to

$$\int_a^b ff(\xi)d\xi \approx \sigma h \left\{ gs'(0)ff(\gamma) + \sum_{n=1}^{N} \omega_n [ff(a + \sigma\delta_n) + ff(b - \sigma\delta_n)] \right\} \tag{44}$$

As with any other quadrature rule, singularities of the integrand near the integration path adversely affect the convergence. However, any singularities on the integration path are easily treated by splitting the integration range so that the singularities are placed at the endpoints [12]. Therefore, the integration path in the first part of Equation (20) should be separated into (45) to ensure the convergence of the integration.

$$\int_0^p (\Gamma_m - \Gamma_m^\infty) J_0(mr)dm = \left(\int_0^{bk} + \int_{bk}^p\right)(\Gamma_m - \Gamma_m^\infty) J_0(mr)dm \tag{45}$$

where breakpoint bk is set to the real part of wavenumber k_i in this paper.

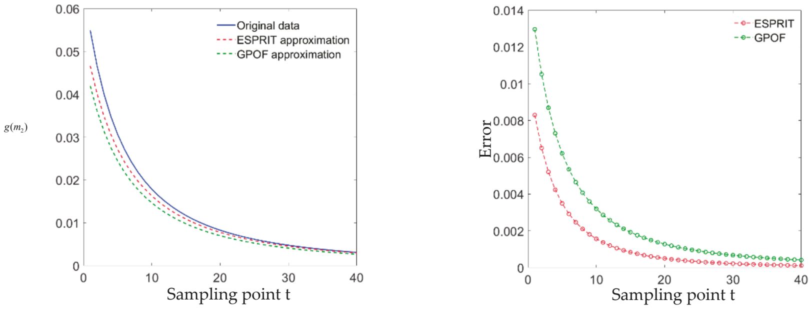
4. Results

In this section, the half-space model is designed to test the correctness of the proposed method by comparing it with DE rules and partition-extrapolation WA algorithm [12], which was well accepted and known as one of the most accurate and efficient ones. Finally, the DE_DCIM method is utilized for the calculation of the Green function of the three-layer model in comparison with the finite element method [24].

4.1. Half-Space

It is assumed that the size of the study area is 1410 m $*$ 1410 m $*$ 710 m; the sampling interval in the horizontal and vertical directions is 10 m, taking the main frequency of 20 Hz Ricker wavelet as the source; and the simulation frequency is 10 Hz (all the study areas in the following are consistent with this study area). Consider a half-space defined by interface z_1 = 350 m, with a point source located at (x_s, y_s, z_s) = (710 m, 710 m, 360 m). Assume that the velocity and density parameters are set as v_1 = 340 m/s, ρ_1 = 0.00129 g/cm^3, v_2 = 2000 m/s, ρ_2 = 1.5 g/cm^3. The integration path is divided into three segments, the breakpoint bk= real$(\omega/\widetilde{v}_i)$, and the breakpoint p is set as $k_i\sqrt{1+(198*5*T_{02}/200)^2}$.

Figure 2 shows the comparison of ESPRIT and GPOF methods to extract DCIM of the lower half-space. $g(m_2)$ is defined by (13) and (14); sampling point t is defined in (23). From Figure 2a, both methods gain a good fit with the original data, but the ESPRIT method is slightly more accurate, with a fitting error less than 0.008, while GPOF [25] is less than 0.014. Further, the time cost in computing half-space DCIM is also presented in Table 1, which shows ESPRIT also reduces the calculation time. The more layers there are, the more obvious time saving will be seen.



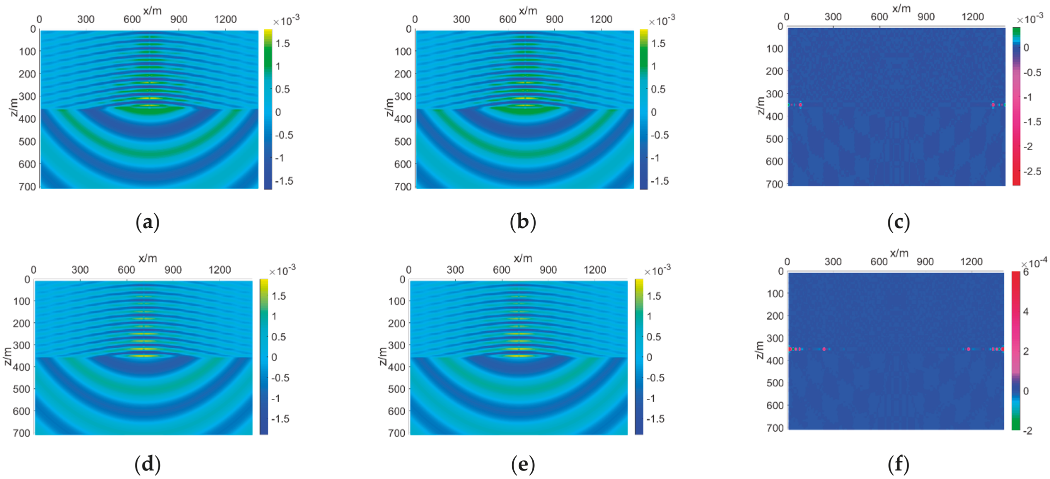
(**a**) (**b**)

Figure 2. Comparison of ESPRIT and GPOF results with original data. (**a**) Fitting results; (**b**) fitting error.

Table 1. Computation time comparison with different methods.

Method	GPOF	ESPRIT
Computation time	0.135 s	0.025 s

Figure 3 shows the symmetrical wavefield of (x, y_s, z) plane. Comparing the solution of DE_DCIM and the numerical integration with DE_WA, the relative errors are shown in (c) and (f). Excellent agreement is obtained with a relative error of real part less than 2.5×10^{-3} and of image part less than 5.8×10^{-4}, which assesses the validity of the present method. The calculation time of the half-space with different parameters is shown in Table 2. It can be seen from Table 2 that the proposed DE_DCIM method reduced the computational time by about 40% when compared to the DE_WA method, with $\rho_1 = \rho_2 = 1.0\ \mathrm{g/cm^3}$.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 3. Comparison of DE_DCIM and DE_WA solution in half-space. (**a**) real part of DE_DCIM solution; (**b**) real part of DE_ WA solution; (**c**) real part of relative error between (**a**,**b**). (**d**) Imaginary part of DE_DCIM solution; (**e**) imaginary part of DE_ WA solution; (**f**) imaginary part of relative error between (**d**,**e**).

Table 2. Computation time comparison with different parameters (computation of (x, y_s, z) plane).

Method\Parameters	v_1 **= 340 m/s** v_2 **= 2000 m/s**	v_1 **= 1000 m/s** v_2 **= 2000 m/s**	v_1 **= 2000 m/s** v_2 **= 3000 m/s**
DE_WA	91.2 s	67.8 s	57.6 s
DE_DCIM	50.8 s	38.9 s	32.5 s
Time saving	44.3%	42.6%	43.6%

4.2. Three-Layer Model

Consider a three-layer structure defined by two interfaces placed at z_1 = 200 m, z_2 = 500 m, with a point source located at (x_s, y_s, z_s) = (710 m, 710 m, 200 m). Assume that the velocity and density parameters are set as $v_1 = 1000\ \mathrm{m/s}$, $\rho_1 = 1.5\ \mathrm{g/cm^3}$, $v_2 = 2000\ \mathrm{m/s}$, $\rho_2 = 2.0\ \mathrm{g/cm^3}$, $v_3 = 3000\ \mathrm{m/s}$, $\rho_3 = 3.0\ \mathrm{g/cm^3}$. The method in this paper is used to solve the layered model, and the symmetrical wavefield, as shown in Figure 4, is obtained. It can be seen from (a) and (d) that in the first layer, there are only up-going wave fields; in the third layer, only down-going wave fields; and in the second layer, there are upward and downward wave fields, which are mixed. Comparing with the FEM [24], the results are consistent in shape, and there are some numerical differences. The relative error of the real part is less than 0.09, and the imaginary part is less than 0.04. Figure 4 also indicates that it is correct to calculate the layered space wave field according to Formulas (11) and (12).

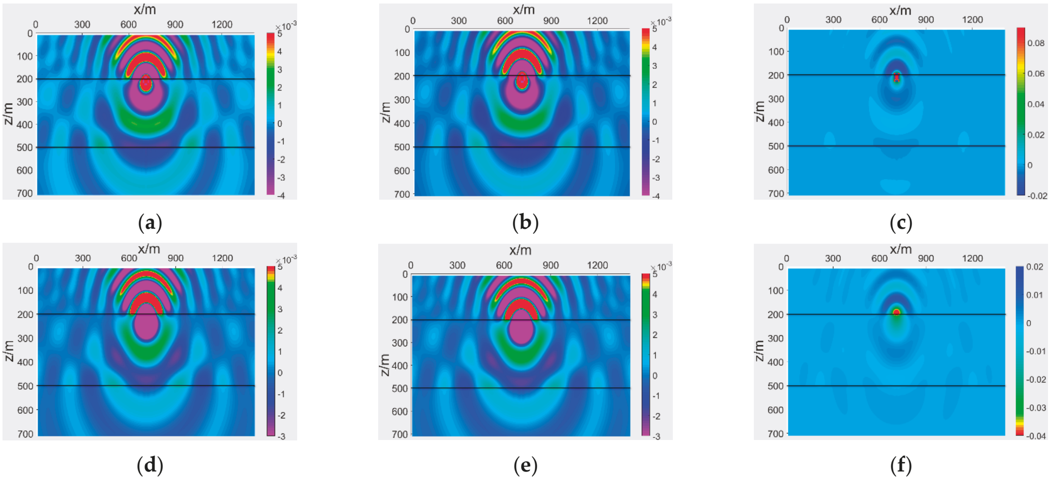


Figure 4. Comparison of the wavefield of 3-layer model (DE_DCIM method and FEM method). (**a**) Real part of DE_DCIM solution; (**b**) real part of FEM solution; (**c**) real part of relative error between (**a**,**b**). (**d**) Imaginary part of DE_DCIM solution; (**e**) imaginary part of FEM solution; (**f**) imaginary part of relative error between (**d**,**e**).

To show the advantage of DE_DCIM over DCIM for accuracy, consider the three-layer structure defined above with a point source located at (x_s, y_s, z_s) = (0 m, 710 m, 200 m). Figure 5 compares the solutions of DE_WA, DE_DCIM, and DCIM in line (x, y_s, z_s). The three-level DCIM with surface wave extraction [26] is adopted. It can be observed in the figure that the DE_DCIM result is more accurate than the three-level DCIM for about two orders of magnitude when both compared to DE_WA, especially when the distance between source and field point is large. As discussed in reference [27], for multilayer media, it is very difficult to find surface wave poles, and the inaccurate extraction of the surface wave will bring unpredictable errors to the results.

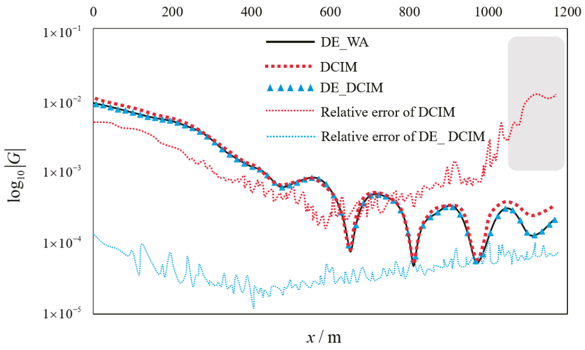


Figure 5. The results of $\log_{10}|G|$ and relative errors in line (x, y_s, z_s) of 3-layer model.

5. Conclusions

Sommerfeld integral is included in the Green function for the seismic field in horizontal layered half-space. The numerical technique is used to compute the Sommerfeld integrals by deriving the integral into two parts, the infinite integral part and the finite integral part, and by applying DE quadrature rules to evaluate the finite part and DCIM to calculate the infinite part. Compared with the DE_WA method, the new method can get an accurate result with a relative error less than 2.5×10^{-3} and increase time saving by about 40%. The

ESPRIT method is introduced to extract DCIM for better accuracy and efficiency. Finally, the fast numerical method is applied to the calculation of the seismic field in horizontal layered half-space. The method in this paper takes both efficiency and accuracy into account, and theoretically, higher accuracy can be achieved by controlling the parameters.

Author Contributions: Conceptualization, Z.Z. and S.D.; methodology, S.L.; software, S.L.; validation, S.L., Y.Y. and Z.Z.; formal analysis, S.L.; writing—original draft preparation, S.L.; writing—review and editing, I.I. and Y.Y.; supervision, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundamental Research Funds for the Central Universities of Central South University 2019ZZTS300.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to some institutional reasons.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fuchs, K.; Müller, G. Computation of Synthetic Seismograms with the Reflectivity Method and Comparison with Observations. *Geophys. J. R. Astron. Soc.* **1971**, *23*, 417–433. [CrossRef]
2. Yao, Z.; Zheng, T. A Generalized Reflection-transmission Coefficient Matrix and Discrete Wavenumber Method for Synthetic Seismograms (ii)- Multiple sources at Different Depths. *Chin. J. Geophys.* **1984**, *27*, 338–348.
3. Apsel, R.J. Dynamic Green's functions for layered media and applications to boundary-value problems. *Aviat. Week Space Technol.* **1979**, *125*, 209–211. [CrossRef]
4. Bouchon, M. A simple method to calculate Green's functions for elastic layered media. *Bull. Seism. Soc. Am.* **1981**, *71*, 959–971. [CrossRef]
5. Olson, A.H. *Forward Simulation and Linear Inversion of Earthquake Ground Motions*; University of California: San Diego, CA, USA, 1982.
6. Yao, Z. A generalized reflection-transmission coefficient matrix and discrete wavenumber method for synthetic seismograms. *Bull. Seism. Soc. Am.* **1983**, *73*, 647–654.
7. Karabulut, E.P.; Aksun, M.I. A Novel approach for the Efficient and Accurate Computation of Sommerfeld Integral Tails. In Proceedings of the 2015 International Conference on Electromagnetics in Advanced Applications (ICEAA), Torino, Italy, 7–11 September 2015.
8. Guo, L. Research and Application of Fast Integral Equation Method Based on Layered Medium Green's Function. Ph.D. Thesis, School of Electronic Engineering of UESTC, Chengdu, China, 2016.
9. Press, W.H.; Flannery, B.P.; Teukolsky, S.A.; Vetterling, W.T.; Chipperfield, J.R. Numerical recipes: The art of scientific computing: Cambridge University Press, Cambridge, 1986 (ISBN 0-521-30811-9). xx + 818 pp. Price 25.00. *Anal. Chim. Acta.* **1987**, *199*, 293–294. [CrossRef]
10. Golubovic, R.; Polimeridis, A.G.; Mosig, J.R. Efficient Algorithms for Computing Sommerfeld Integral Tails. *IEEE Trans. Antennas Propag.* **2012**, *60*, 2409–2417. [CrossRef]
11. Niciforovic, R.G.; Polimeridis, A.; Mosig, J.R. Fast Computation of Sommerfeld Integral Tails via Direct Integration Based on Double Exponential-Type Quadrature Formulas. *IEEE Trans. Antennas Propag.* **2011**, *59*, 694–699. [CrossRef]
12. Michalski, K.A.; Mosig, J.R. Efficient computation of Sommerfeld integral tails—Methods and algorithms. *J. Electromagn. Waves Appl.* **2016**, *30*, 281–317. [CrossRef]
13. Mosig, J. The Weighted Averages Algorithm Revisited. *IEEE Trans. Antennas Propag.* **2012**, *60*, 2011–2018. [CrossRef]
14. Michalski, K.A. Extrapolation methods for Sommerfeld integral tails. *IEEE Trans. Antennas Propag.* **1998**, *46*, 1405–1418. [CrossRef]
15. Mosig, J.R.; Gardiol, F.E. A Dynamical Radiation Model for Microstrip Structures. *Adv. Electron. Electron Phys.* **1982**, *59*, 139–237.
16. Takahasi, H.; Mori, M. Double Exponential Formulas for Numerical Integration. *Publ. Res. Inst. Math. Sci.* **1973**, *9*, 721–741. [CrossRef]
17. Rao, Y.; Wang, Y. Fracture effects in seismic attenuation images reconstructed by waveform tomography. *Geophysics.* **2009**, *74*, 25. [CrossRef]
18. Malinowski, M.; Operto, S.; Ribodetti, A. High-resolution seismic attenuation imaging from wide-aperture onshore data by visco-acoustic frequency-domain full-waveform inversion. *Geophys. J. Int.* **2011**, *186*, 1179–1204. [CrossRef]
19. Carpio, A.; Rapún, M.L. Multifrequency Topological Derivative Approach to Inverse Scattering Problems in Attenuating Media. *Symmetry* **2021**, *13*, 1702. [CrossRef]

20. Li, Q. *The Way to Precise Exploration: Systematic Engineering Analysis of High Resolution Seismic Exploration*; Petroleum Industry Press: Beijing, China, 1994.
21. Touhei, T. A scattering problem by means of the spectral representation of Green's function for a layered acoustic half-space. *Comput. Mech.* **2000**, *25*, 477–488. [CrossRef]
22. Faisal, K.M. Improved Matrix Pencil Methods for Parameters Estimation of Plane Wave Signals. Ph.D. Thesis, Department of Electrical Engineering, Pakistan Institute of Engineering & Applied Sciences (PIEAS), Nilore, Islamabad, Pakistan, 2011.
23. Zhang, X. *Modern Signal Processing*, 2nd ed.; Tsinghua University Press: Beijing, China, 2002.
24. Xu, K.; Wang, M. Parameter inversion of acoustic wave equation in frequency domain with Finite element method. *Chin. J. Geophys.* **2001**, *44*, 852–864. [CrossRef]
25. Hua, Y.; Sarkar, T.K. Generalized pencil-of-function method for extracting poles of an EM system from its transient response. *IEEE Trans Antennas Propagation.* **1989**, *37*, 229–234. [CrossRef]
26. Alparslan, A.; Aksun, M.I.; Michalski, K.A. Closed-Form Green's Functions in Planar Layered Media for All Ranges and Materials. *IEEE Trans. Microw. Theory Tech.* **2010**, *58*, 602–613. [CrossRef]
27. Boix, R.R.; Fructos, A.L.; Mesa, F. Closed-Form Uniform Asymptotic Expansions of Green's Functions in Layered Media. *IEEE Trans. Antennas Propag.* **2010**, *58*, 2934–2945. [CrossRef]

applied sciences

MDPI

Article

Design of Gas Cyclone Using Hybrid Particle Swarm Optimization Algorithm

Xueli Shen and Daniel C. Ihenacho *

Department of Software Engineering, Liaoning Technical University, Huludao 125000, China; Shenxueli@lntu.edu.cn
* Correspondence: 201909019@stu.lntu.edu.cn

Abstract: The method of searching for an optimal solution inspired by nature is referred to as particle swarm optimization. Differential evolution is a simple but effective EA for global optimization since it has demonstrated strong convergence qualities and is relatively straightforward to comprehend. The primary concerns of design engineers are that the traditional technique used in the design process of a gas cyclone utilizes complex mathematical formulas and a sensitivity approach to obtain relevant optimal design parameters. The motivation of this research effort is based on the desire to simplify complex mathematical models and the sensitivity approach for gas cyclone design with the use of an objective function, which is of the minimization type. The process makes use of the initial population generated by the DE algorithm, and the stopping criterion of DE is set as the fitness value. When the fitness value is not less than the current global best, the DE population is taken over by PSO. For each iteration, the new velocity and position are updated in every generation until the optimal solution is achieved. When using PSO independently, the adoption of a hybridised particle swarm optimization method for the design of an optimum gas cyclone produced better results, with an overall efficiency of 0.70, and with a low cost at the rate of 230 cost/s.

Keywords: particle swarm optimization (PSO); differential evolution (DE); gas cyclone; hybridised particle swarm optimization; evolutionary algorithm

Citation: Shen, X.; Ihenacho, D.C. Design of Gas Cyclone Using Hybrid Particle Swarm Optimization Algorithm. *Appl. Sci.* **2021**, *11*, 9772. https://doi.org/10.3390/app11209772

Academic Editors: Peng-Yeng Yin, Ray-I Chang, Youcef Gheraibia, Ming-Chin Chuang, Hua-Yi Lin and Jen-Chun Lee

Received: 4 September 2021
Accepted: 12 October 2021
Published: 19 October 2021

1. Introduction

Particle swarm optimization (PSO) and differential evolution (DE) are two stochastic, population-based optimization EAs in evolutionary algorithms.

PSO was developed by Kennedy and Eberhart and was originally intended to simulate social behaviour. Every solution in PSO is a "bird" in the search space [1]. However, when the technique is implemented, it is referred to as a particle. All of the particles have fitness values that the fitness function evaluates in order to optimize them, as well as velocities that control their flight. The particles navigate through the problem space by following the current best particles. Although the PSO algorithm has attracted a lot of attention in the last decade, it unfortunately has a premature convergence issue, which is common in complicated optimization problems. To improve PSO's search performance, certain strategies for adjusting parameters such as inertia weights and acceleration coefficients have been developed.

Storn and Price [2] presented DE as a simple but effective EA for global optimization. The DE method has progressively gained popularity and has been utilized in a variety of practical applications, owing to its shown strong convergence qualities and ease of understanding [3]. DE has been successfully applied in a variety of engineering disciplines [4–8]. The selected trial vector generation technique and related parameter values have a significant impact on the performance of the traditional DE algorithm. Premature convergence can also be caused by poor methodology and parameter selection. DE scholars have proposed a number of empirical guidelines and proposals for selecting trial vector generation techniques and their associated control parameter settings during the last decade [9–12].

Despite the fact that PSO has been successfully applied to a wide range of challenges, including test and real-world scenarios, it has faults that might cause the algorithm performance to suffer. The major problem is a lack of diversity, which leads to a poor solution or a slow convergence rate [13]. The hybridization of algorithms, in which two algorithms are combined to generate a new algorithm, is one of the groups of modified algorithms used to improve the performance. DE is applied to each particle for a certain number of iterations in order to choose the best particle, which is then added to the population [14,15]. The barebones DE [16] is a proposed hybrid version of PSO and DE. The evolving candidate solution is created using DE or PSO and is based on a specified probability distribution [17]. In a hybrid metaheuristic [13], the strengths of both techniques are retained.

Cyclone separators are a low-cost and low-maintenance way of separating particulates from air streams. A cyclone is made up of two parts: an upper cylindrical element called the barrel and a lower conical part called the cone, as seen in Figure 1. The air stream enters the barrel tangentially and goes downhill into the cone, generating an outer vortex. The particles are separated from the air stream by a centrifugal force caused by the increased air velocity in the outer vortex. When the air reaches the bottom of the cone, an inner vortex forms, reversing the direction of the air and exiting out the top as clean air, while particulates fall into the dust collection chamber attached to the bottom of the cyclone.

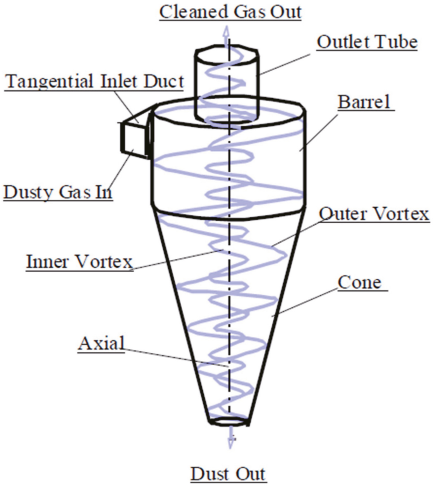


Figure 1. Schematic flow diagram of a cyclone [18].

The simulation of an optimum gas cyclone with low cost is the primary focus of this research with the use of PSO, DE, and hybrid DEPSO algorithms using an objective function which is of a minimization type. The efficient global optimization is a major advantage of the DE algorithm. Furthermore, the diversity of the entire population is easily maintained throughout the process, preventing individuals from falling into a local optimum. PSO, on the other hand, has the advantage of a quick convergence speed. The best individual particle across the entire iteration is saved to obtain the lowest fitness values. Combining the benefits of DE and PSO, the use of a hybrid DEPSO strategy is proposed for this research with the goal of fast convergence and efficient global optimization. The proposed DEPSO method first reduces the search space using the DE algorithm, and then the obtained populations are used as the initial population by the PSO to achieve a fast convergence rate to a final global optimum. Moreover, DEPSO utilizes the crossover operator feature of DE to improve the distribution of information between candidates, and based on the fitness function, the hybrid algorithm can determine the global minimum cost value much better than the use of PSO alone due to its drawbacks, such as high computational complexity, slow convergence, sensitivity to parameters, and so forth.

The overview of this paper is as follows:

- Sections 2 and 3 describe the PSO and DE algorithms, respectively.
- Section 4 provides insight about the DESPO algorithm.
- Section 5 gives a brief description about the gas cyclone device's operation.
- Section 6 consists of the experimental results and performance evaluation of each algorithm.
- Finally, Section 7 concludes the research work.

2. PSO

The theory behind particle swarm intelligence (PSO) is primarily nature-inspired, emulating the behaviour of animal societies by following the member of the group that is closest to the food source, such as flocks of birds and schools of fish. For example, a flock of birds during food search will follow a member of the flock that is closest to the position of the food source (the potential solution). This is achieved because of the simultaneous interactions made with each member of the flock in search of the best position. This continues until the food source is discovered (the best solution).

The PSO algorithm follows this process in search of the best solution to a given problem. The algorithm consists of a swarm of particles in which each particle is a potential solution, which usually leads to the best solution. This best solution is known as the global best and is called $g(t)$ or g_{best}.

After obtaining the two best values, the particle updates its velocity and positions with Equations (1) and (2):

$$v_i(t+1) = wv_i(t) + r_1c_1(P_i(t) - x_i(t)) + r_2c_2(g(t) - x_i(t)) \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

where:

w is inertia;
$v_i(t)$ is the particle velocity;
$x_i(t)$ is the current particle (solution);
$P_i(t)$ and $g(t)$ are defined as personal best and global best, respectively;
r_1r_2 are random numbers between (0,1);
c_1c_2 are learning factors;
i is the i-th particle.

The pseudo-code (Algorithm 1) and the flowchart (see Figure 2) of the procedure are shown below.

Algorithm 1 PSO

```
Input: Objective function: $f_i$; lower bound: lb; upper bound: ub; population size: $N_p$; velocity: v; dimension size: D; termination criterion: T; inertia weight: w; learning rates: $c_1$ and $c_2$.
Output: Return $g_{best}$ as the best estimation of the global optimum
Initialize the controlling parameters
Population = Initialize Population ($N_p$,D,ub,lb,v)
Evaluate the objective function, $f_i$
Assign $P_{best}$ as Population and $f_{pbest}$ as $f_i$
Identify the solution with the best fitness and assign that solution as $g_{best}$ and fitness as $f_{gbest}$
for t = 0 to T
for $i$ = 0 to $N_p$
```

```
        Calculate the velocity, $v_i$ of $i$-th particle
        Calculate the new position, $X_i$ of $i$-th particle
        Bound $X_i$
        Evaluate objective function $f_i$ of $i$-th particle
        Update population using $X_i$ and $f_i$
        if ($f_i < f_{pbest,i}$) then
            $P_{best,i} = X_i$
            $f_{best,i =}\ f_i$
        end if
        if ($f_{Pbest,i} < f_{gbest}$) then
          $g_{best} = P_{best,i}$
          $f_{gbest\,=}\ f_{pbest,i}$
        end if
        Update Inertia, w
    end for
    end for
```

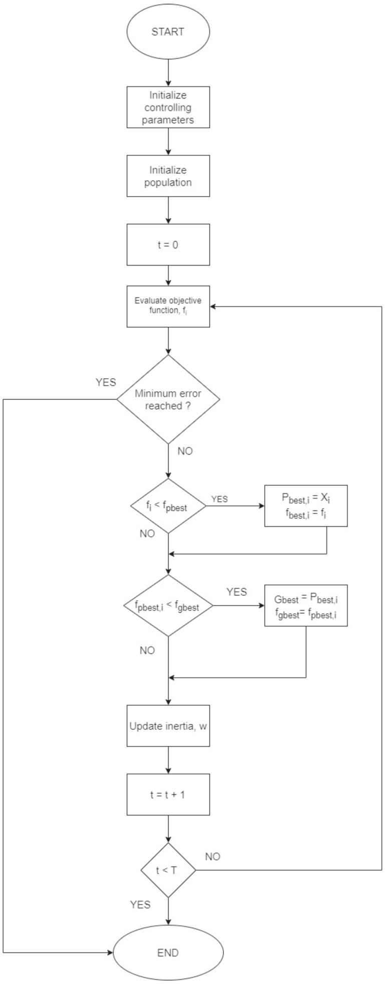


Figure 2. Flowchart of PSO algorithm.

3. DE

Similar to other evolutionary computation methods, DE starts with an initial population that is initialized randomly. After determining the population, a new candidate individual is generated by applying mutation and crossover operators. This candidate then becomes the input of the selection operator, and through a hard selection mechanism between the candidate and the current member of the population, if the candidate is better than the current member, it will enter the next generation; otherwise, the current member remains in the population. Algorithm 2 is the pseudo-code of the standard DE. In the algorithm, N_P denotes the population size, D is the dimension size, F is the scaling factor, C_r is the crossover rate, randint (1, D) is the random integer [1, D], and rand is the random real number in [0, 1]. The main operator in DE is the mutation operator. These mutations take advantage of diverse mechanisms and strategies for generating the *donor* vector. In order to distinguish between DE variants, the notation DE/X/Y/Z is proposed by Storn and Price, where DE denotes differential evolution, X denotes the base vector, Y denotes number of difference vectors, and finally Z determines the type of crossover.

The pseudo-code and flowchart (see Figure 3) of the procedure is as follows.

Algorithm 2 Pseudo-code for classic differential evolution

```
Inputs: Fitness function: f; lb: lower bound; ub: upper bound: $N_p$: population size; T: termination
criteria; F: scaling factor/mutation rate; $P_c/C_r$: crossover probability; D: dimension size.
Output: Best_Vector
Population = Initialize Population ($N_p$,D,ub,lb)
while (T $\neq$ True) do
    Best_Vector = Evaluate_Population (Population)
    $V_x$ = Select_Random_Vector (Population)
Index = Find_Index_of_Vector($V_x$) //Specify row number of a vector
    Select_Random_Vector (Population, $v_1$,$v_2$,$v_3$)//where $v_1 \neq v_2 \neq v_3 \neq v_x$
    $V_y = v_1 + F (v_2 - v_3)$// donor vector
    for ($i$ = 0;$i$ + +;I < D – 1)//Loop for starting Crossover
operation
        if (rand$_j$ [0,1] < $C_r$) then
            u[$i$] = $v_x$[$i$]
        else u[$i$] = $v_y$[$i$]
    end for //end crossover operation
        if (f(u) $\leq$ f($v_x$)) then
            Update Population (u, Index, Population)
end//While loop
return Best_Vector
```

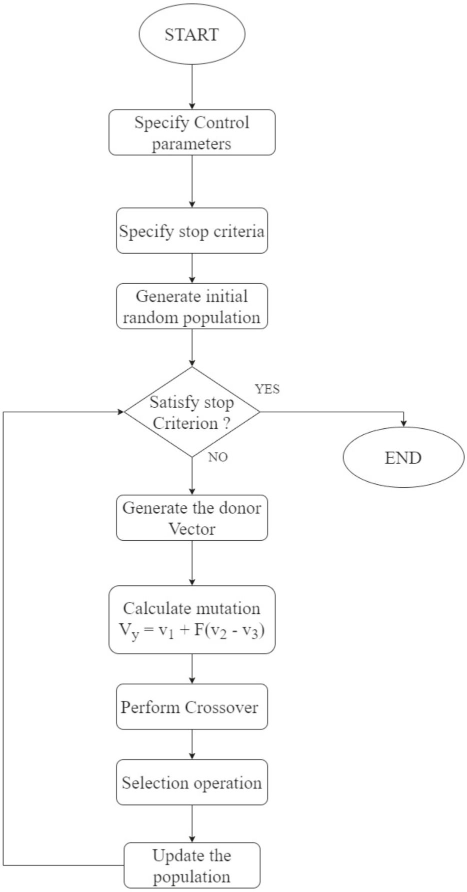

START
Specify Control parameters
Specify stop criteria
Generate initial random population
Satisfy stop Criterion ?
YES
NO
END
Generate the donor Vector
Calculate mutation $V_y = v_1 + F(v_2 - v_3)$
Perform Crossover
Selection operation
Update the population

Figure 3. Flowchart of classic differential evolution.

4. Hybridised PSO

Hybridised PSO is a method that combines PSO with various classical and evolutionary optimization algorithms to make use of the strengths of both approaches while compensating for their flaws. DEPSO, a hybridised PSO which combines DE and PSO, is the suggested hybridised PSO for this research.

DEPSO follows the same steps as the standard DE method until the trial vector is created. If the trial vector meets the criteria, it is added to the population; otherwise, the algorithm moves on to the PSO phase and creates a new candidate solution. Iteratively, the technique is repeated until the optimal value is found. The incorporation of the PSO phase generates a disturbance in the population, which aids in population diversification and the output of an optimum solution. The following is an illustration of the algorithm's procedure [19]:

1. *Population initialization*: The individual x, with the population number N_P, is randomly generated to form an initial population in a D-dimensional space. All the individuals should be generated within the bounds of the solution space. The initial individuals are generated randomly in the range of the search space. Additionally, the associated velocities of all particles in the population are generated randomly in the D-dimension space. Therefore, the initial individuals and the initial velocity can be expressed as follows [19]:

$$\begin{aligned} X_i(0) &= \{x_{i,1}(0), x_{i,2}(0), \ldots, x_{i,D}(0)\} \\ x_{i,j}(0) &= x_{min} + rand_{i,j}(0,1) \times (x_{max} - x_{min}) \\ V_i(0) &= \left\{v_{i,1}(0), v_{i,2}, \ldots, v_{i,N_p}(0)\right\}, \end{aligned} \tag{3}$$

where $1 \le i \le N_p \le j \le$ D, $[x_{min}, x_{max}]$ is the range of the search space, and rand(0,1) is a random number chosen between 0 and 1.

2. *Iteration Loop of DE:* The individual mutation operation is denoted as time *t*. By randomly choosing three individuals from the previous population, the mutant individual (*t* + 1) can be generated as [19]:

$$V_i\ (t+1) = X_{r^1}(t) + F(X_{r^2}\ (t) - X_{r^3}\ (t)) \tag{4}$$

where *F* is a differential weight between 0 and 1.

The crossover operation aims to construct a new population $u_{i,}(t+1)$, which is chosen from the current individuals and mutant individuals in order to increase the diversity of the generated individuals [19]:

$$\begin{aligned} U_i\ (t\ +\ 1)\ &=\ \{u_{i,1}\ (t\ +\ 1)\ ,\ u_{i,2}\ (t\ +\ 1)\ ,\cdots,\ u_{i,D}\ (t\ +\ 1)\} \\ u_{i,j}(t+1) &\begin{cases} v_i(t+1),\ if\ rand\ (0,1)\ \le C_r\ or\ j\ =\ j_{rand} \\ x_i(t),\ otherwise, \end{cases} \end{aligned} \tag{5}$$

where rand(0, 1) is a random number chosen from 0 to 1, j_{rand} is an integer chosen from 1 to *D* randomly, and C_r is a crossover parameter that is randomly chosen from 0 to 1.

In the selection operation, the crossover vector *U*(*t*+1) is compared to the target vector $X_i(t)$ by evaluating the fitness function value based on a greedy criterion, and the vector with a smaller fitness value is selected as the next generation vector:

$$X_i(t+1) = \begin{cases} U_i(t+1),\ if\ f(X_i(t))\ \ge\ f(U_i(t+1)) \\ X_i(t),\ otherwise, \end{cases} \tag{6}$$

The global best part is updated with the minimum fitness value (*gbest*) and the personal-best part (*pbest*).

3. *Iteration Loop of PSO*: The velocity and position equations of individuals remains unchanged and follows the earlier stated Equations of (1) and (2) respectively in page 3 as shown below:

$$v_i(t+1) = wv_i(t) + r_1c_1(P_i(t) - x_i(t)) + r_2c_2(g(t) - x_i(t))$$
$$x_i(t+1) = x_i(t) + v_i(t+1)$$

The pseudo code (Algorithm 3) and flowchart (see Figure 4) of DEPSO are shown below.

Algorithm 3 DEPSO

```
Inputs: Fitness function: f; lb: lower bound; ub: upper bound; $N_p$: population size; T: termination criteria; F: scaling factor/mutation rate; $P_c/C_r$: crossover probability; D: dimension size; w: inertia weight; learning rates: $c_1$ and $c_2$.
Output: BestVector
Population = Initialize Population ($N_p$,D,ub,lb)
while (T $\neq$ True) do
    Best_Vector = Evaluate_Population (Population)
    $V_x$ = Select_Random_Vector (Population)
Index = Find_Index_of_Vector($V_x$) //Specify row number of a vector
    Select_Random_Vector (Population, $v_1$,$v_2$,$v_3$)//where $v_1 \neq v_2 \neq v_3 \neq v_x$
    $V_y = v_1 + F (v_2 - v_3)$//donor vector
    for ($i$ = 0; $i$ + +; I < D − 1)//Loop for starting Crossover
operation
        if (rand$_j$[0,1] < $C_r$) then
            u[$i$] = $v_x$[$i$]
        else u[$i$] = $v_y$[$i$]
  end for //end crossover operation
        if (f(u) $\leq$ f($v_x$)) then
            Update Population (u, Index, Population)
      else
for $i$ = 0 to $N_p$
    Calculate the velocity, $v_i$ of $i$-th particle
    Calculate the new position, $X_i$ of $i$-th particle
    Bound $X_i$
    Evaluate objective function $f_i$ of $i$-th particle
    Update population using $X_i$ and $f_i$
    if ($f_i < f_{pbest,i}$) then
        $P_{best,i} = X_i$
        $f_{best,i} = f_i$
    end if
    if ($f_{Pbest,i} < f_{gbest}$) then
    $g_{best} = P_{best,i}$
    $f_{gbest} = f_{pbest,i}$
     Update Population (u, Index, Population)
    end if
    Update Inertia, w
end for
end//While loop
return BestVector
```

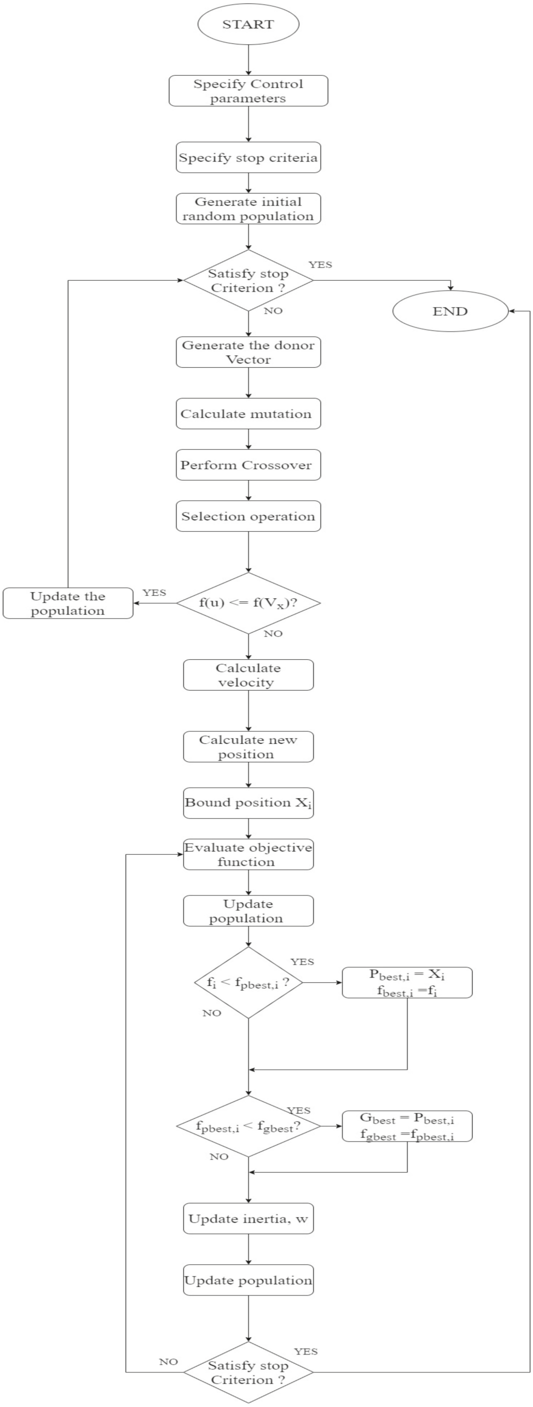


Figure 4. Flowchart of DEPSO.

5. Gas Cyclone Design

Cyclones are devices used for sizing, classification, and screening of particulate materials in mixtures with fluid (gases or liquid). Cyclones come in many sizes and shapes and have no moving parts. The mode of operation involves the process of subjecting the flowing fluid to swirl around the cylindrical part of the device. They impact the cyclone walls, fall down the cyclone wall (by gravity), and are collected in a hopper. The most important parameter of a cyclone is its collection efficiency and the pressure drop across the unit. Cyclone efficiency is increased through:

(a) Reduction in cyclone diameter, gas outlet diameter, and cone angle;
(b) Increasing the cyclone body length.

Capacity is, however, improved by increasing the cyclone diameter, inlet diameter, and body length. Increasing the pressure drop give rise to:

- Increase in separation efficiency;
- Higher capacity;
- Decrease in the underflow to throughput ratio;
- Cleaner overflow.

The design parameters include:

- Feed rate Q m^3/s;
- Cyclone diameter D_c, m;
- Set cut size d_{pc}, μm;
- Cyclone efficiency;
- Cyclone pressure drop, D_P;
- Cyclone geometry (D (cyclone diameter), De(vortex finder diameter), h(cylindrical height), H(overall height), a(inlet height), b(inlet width), B(cone outlet diameter), S(vortex finder height));
- Temperature, °C;
- Pressure, N/m^2;
- Viscosity, Ns/m^2;
- Density fluid, kg/m^3;
- Mass median diameter MMD, (μm);
- Geometric standard deviation GSD, μm;
- Number of cyclones;
- Fluid density, kg/m^3;
- Particle density, kg/m^3.

Given the cyclone geometry (as shown in Figure 5) and the operating conditions, there are five design parameters that can be specified for a design.

These parameters are:

1. Feed rate, Q;
2. Cyclone diameter, D;
3. Set cut size, d_{pc};
4. Cyclone efficiency;
5. Cyclone pressure drop, D_P.

Cyclone geometry:

- a is the inlet height
- b is the inlet width
- D is the cyclone diameter
- De is the vortex finder diameter
- h is the cylindrical height
- B is the cone diameter
- H is the overall heightIt can be seen from
- S is the vortex finder height.

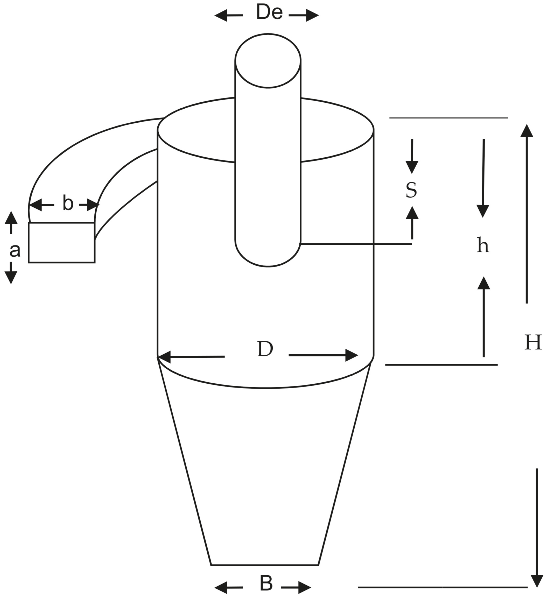


Figure 5. The gas cyclone geometry.

5.1. Particle Cut-Off Size D_{PC}

According to Wang et al. [20–22], cyclone performance depends on the geometry and operating parameters of the cyclone, as well as the particle size distribution of the entrained particulate matter. Several mathematical models have been developed to predict cyclone performance. Lapple [23] developed a semi-empirical relationship to predict the cut point of cyclones designed according to the classical cyclone design method, where cyclone cut point is defined as the particle diameter corresponding to a 50% collection efficiency. Wang et al. showed that Lapple's approach did not discuss the effects of particle size distribution on cyclone performance. The Lapple model was based on the terminal velocity of particles in a cyclone [23]. From the theoretical analysis, Equation (7) was derived to determine the smallest particle that will be collected by a cyclone if it enters at the inside edge of the inlet duct:

$$d_{pc} = \sqrt{\frac{9\mu b}{2\Pi N_e V_i(\rho p - \rho g)}} \tag{7}$$

where:

d_{pc} (cut-point) = diameter of the smallest particle that will be collected by the cyclone if it enters on the inside edge of the inlet duct (μm);
μ = gas viscosity (kg/m/s);
b = width of inlet duct (m);
Ne = number of turns of the air stream in the cyclone;
V_i = gas inlet velocity (m/s);
ρp = particle density (kg/m^3);
ρg = gas density (kg/m^3).

5.2. Fractional Efficiency Calculation

The most important parameters in cyclone operation are pressure drop and collection efficiency. The pressure drop is given by the difference between the static pressure at the cyclone entry and the exit tube. The fractional efficiency for the *j*-th particle size, according to Lapple, is given as [23]:

$$\eta_j = \frac{1}{1+\sqrt{\frac{dpc}{dpj}}} \tag{8}$$

where:

d_{pc} = diameter of the smallest particle that will be collected by the cyclone with 50% efficiency,

d_{pj} = diameter of the *j*-th particle.

The overall collection efficiency of the cyclone is a weighted average of the collection efficiencies for the various size ranges, namely:

$$\eta = \sum \frac{\eta jmj}{m} \tag{9}$$

where:

η = overall collection efficiency;
η_j = fractional efficiency for *j*-th particle size;
m = total mass of particle;
mj = mass of particle in the *j*-th particle size range.

5.3. Pressure Drop Calculation

The energy consumed in a cyclone is most frequently expressed as the pressure drop across the cyclone. This pressure drop is the difference between the gas static pressure measured at the inlet and outlet of the cyclone. Many models have been developed to determine this pressure drop. Some of the commonly used equations to calculate the pressure drop are:

(a) The Koch and Licht Pressure Drop Equation.

Koch and Licht (1977) expressed the cyclone pressure drop as

$$\Delta \mathrm{P} = 0.003\rho_{\mathrm{g}} \mathrm{V}_{i^2}\, \mathrm{N}_{\Pi} \tag{10}$$

where:

ρ_{g} = gas density (lbm/ft^3);
V_i = inlet velocity (ft/s);
N_{Π} = number of velocity heads (inches of water) and is expressed as

$$\mathrm{N}_{\Pi} = \mathrm{K}\left(\frac{a \cdot b}{D_e^2}\right) \tag{11}$$

K = 16 for no inlet vane 7.5 with neutral inlet vane;
a, b = inlet height and width, respectively.

(b) The Ogawa Equation.

Another pressure drop equation due to Ogawa (1984) takes the form of

$$\Delta \mathrm{P} = \frac{1}{2}\rho_{\mathrm{g}} \mathrm{V}_i \mathrm{N}_{\Pi} \tag{12}$$

The prediction of the performance of the cyclone separators is a challenging problem for the designers owing to the complexity of the internal aerodynamic process and dust particles. Hence, modern numerical simulations are needed to solve this problem. Fluid flows have long been mathematically described by a set of nonlinear, partial differential equations, namely the Navier–Stokes equations.

A cyclone is required to remove carbon dust particles from affluent air coming from a thermal power station at a rate of 5.0 m^3/s at 65 °C. The dust particles are assumed to have a normal size distribution with an MMD of 20 µm and a GSD of 1.5 µm. For energy cost consideration, the pressure drop in the cyclone is required to be not more than 1000 N/m^2. The density of the dust particle is 2250 kg/m^3, and the cyclone is operating at atmospheric pressure. To attain overall efficiency of 70% and above, the recommended required geometry/size and cut size of the cyclone are: Q = 5 m^3/s, temp = 65 °C, MMD = 20 µm, GSD = 15 µm, pressure 1000 N/m^2, and ρ dust particle = 2250 kg/m^2.

This study develops PSO, DE, and DEPSO models for the design optimization of a 5.0 m^3/s gas cyclone instrument.

6. Results and Discussion

The cost of the operation of a gas cyclone is the primary problem set for this research, instead of using two conflicting objective functions, namely efficiency and pressure drop. In general, the total cost per unit will be a function of a fixed cost cyclone and the energy cost of operating the cyclone:

C_t = Fixed cost + energy cost.

Assuming the cost of the cyclone depends on its diameter, the fixed cost can be expressed as

$$C_{fixed\ =}\ fN_eDc^2/YH \tag{13}$$

where:

f is an investment factor to allow for installation;

D_c is the cyclone diameter;

N_e is number of turns of the air stream in the cyclone;

H is the time worked per year;

Y is the number of years.

The energy cost is given by

$$C_{energy} = Q\Delta PC_e \tag{14}$$

where:

Q is the feed rate (m^3/s);

ΔP is the pressure drop (Pa);

C_e is cost per unit energy.

Hence,

$$C_t = fN_e\beta_1/YH + \rho_f\ \varepsilon Q^3\ \beta_2/2a_o^2\ b_o^2N^2 \tag{15}$$

where:

$\beta_1 = [d_{pc}{}^2(\rho_s - \rho_f)_{\Pi}N_tN_eQ/9a_ob_o{}^2\ \mu N]^{2/3}$;

$\beta_2 = [d_{pc}{}^2(\rho_s - \rho_f)_{\Pi}N_tQ/9a_ob_o{}^2\ \mu N]^{-4/3}$.

The objective function is of a minimization type, and the EC methods search for an optimized cyclone geometry with low cost per unit. Using the table of parameters as shown in Tables 1–9, the cost performance and overall efficiency of each of the algorithms are shown in the figures below.

The following can be observed from the plots and tables below:

1. It can be seen from Table 10 that DEPSO at 1000 iterations with a particle size of 40 has a cost-effective value of 230 naira/s, with an overall efficiency of 0.70.
2. It can also be seen from Table 10 and Figures 6–19 that the PSO algorithm fails to provide any reasonable results when applied to complex optimization problems, due to its premature convergence, high computational complexity, slow convergence, sensitivity to parameters, and so forth. DEPSO solves this issue by utilizing the crossover operator of DE to improve the distribution of information between candidate solutions.
3. In terms of the best efficiency at low cost for DE algorithm, at a particle size of 60 at 400 iterations, its cost and overall efficiency are 229.365 naira/s and 0.749, respectively, as shown in Figures 10 and 11.

4. In terms of the best efficiency at a low cost for the DEPSO algorithm, at a particle size of 70 at 1000 iterations, its cost and overall efficiency are 249.192 naira/s and 0.786, respectively, as shown in Figures 12 and 13.
5. As seen from Table 10, the DE algorithm's lowest cost is 218.938 naira/s with an overall efficiency of 0.716.

Table 1. Cyclone input data.

Parameters	Value
Feed rate, Q	5.0
Particle size 1	0.00002
Particle size 2	0.0000015
Particle Density	2250
Viscosity	0.00002039
Delta pressure	1000
Particle mass 1	0.000056
Particle mass 2	0.000042
Number of particle range	1
Assumed natural length	2.078

Table 2. Investment/cost data.

Parameters	Value
Investment factor	3.9
Number of years	10
Period of operation/year, (seconds)	0.0000015
Cost/square m of cyclone	20.5
Number of cyclone	1

Table 3. Parameters list with 40-particle size.

Parameters	Value
Particle size	40
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
w	0.729
c_1	2
c_2	2

Table 4. Parameters list with 50-particle size.

Parameters	Value
Particle size	50
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
w	0.729
c_1	2
c_2	2

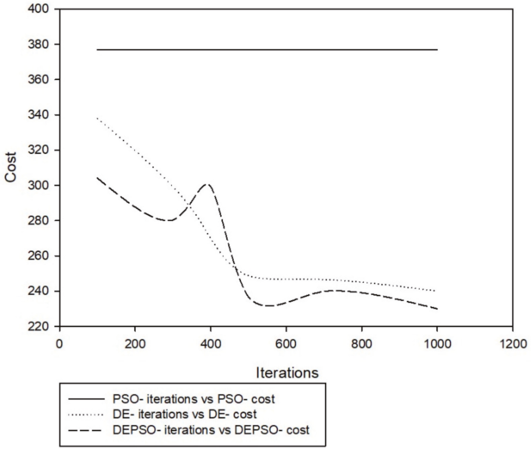


Figure 6. Cost performance of DE, DEPSO, and PSO algorithms at 40 particles.

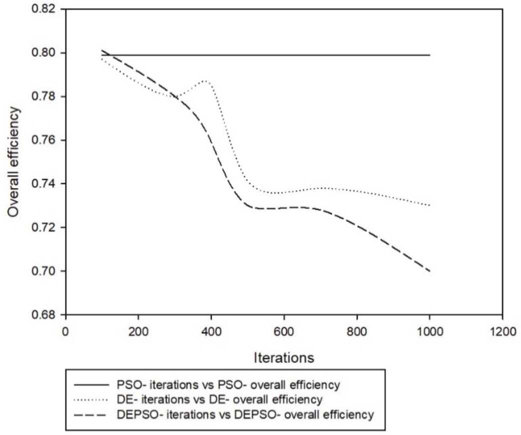


Figure 7. Overall efficiency of DE, DEPSO, and PSO algorithms at 40 particles.

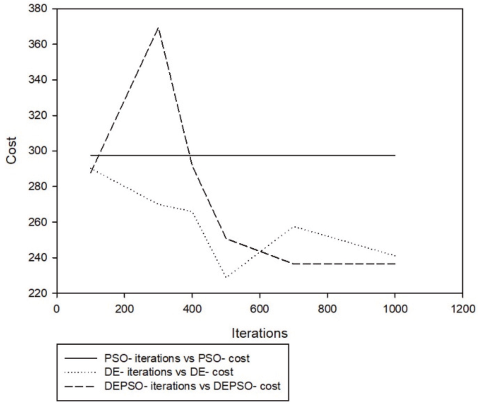


Figure 8. Cost performance of DE, DEPSO, and PSO algorithms at 50 particles.

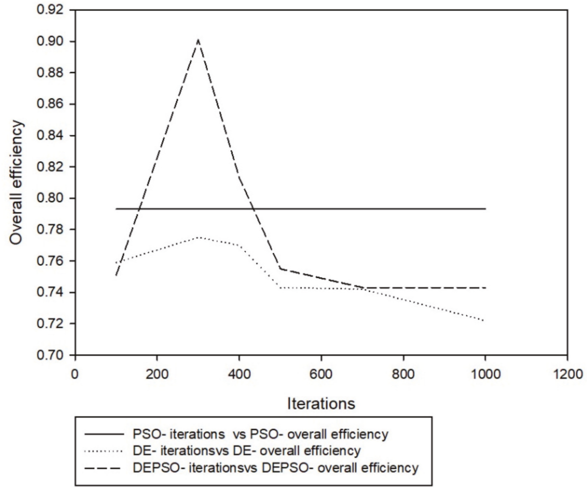


Figure 9. Overall efficiency of DE, DEPSO, and PSO algorithms at 50 particles.

Table 5. Parameters list with 60-particle size.

Parameters	Value
Particle size	60
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
W	0.729
c_1	2
c_2	2

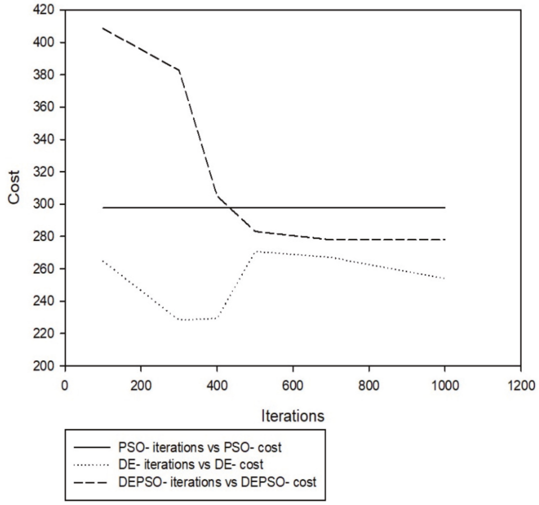


Figure 10. Cost performance of DE, DEPSO, and PSO algorithms at 60 particles.

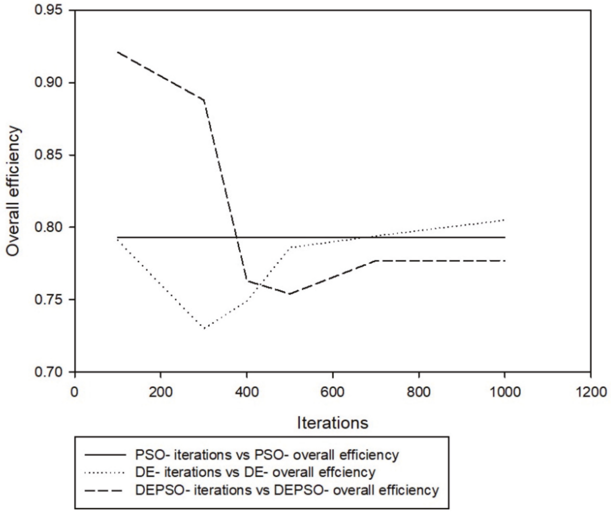


Figure 11. Overall efficiency of DE, DEPSO, and PSO algorithms at 60 particles.

Table 6. Parameters list with 70-particle size.

Parameters	Value
Particle size	70
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
w	0.729
c_1	2
c_2	2

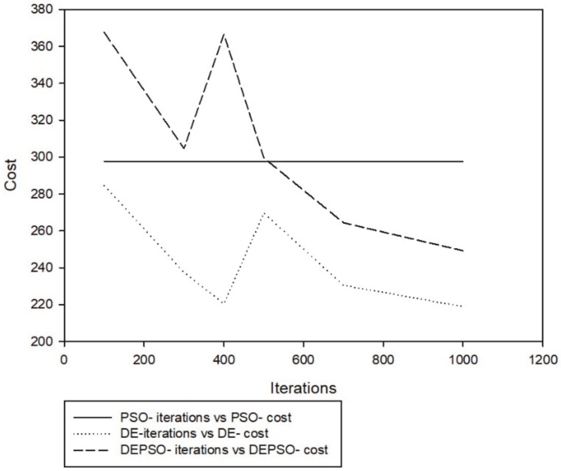


Figure 12. Cost performance of DE, DEPSO, and PSO algorithms at 70 particles.

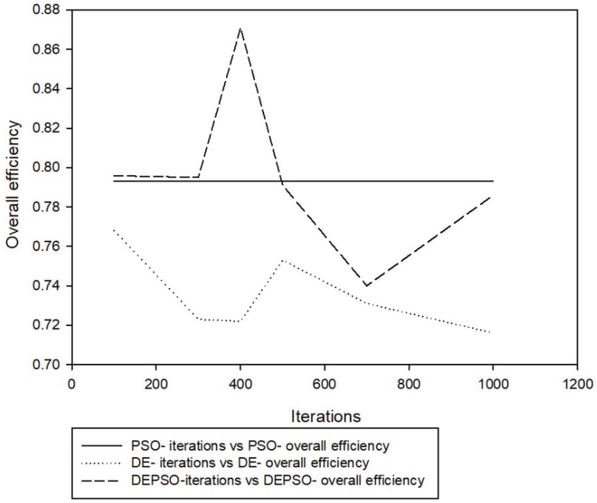


Figure 13. Overall efficiency of DE, DEPSO, and PSO algorithms at 70 particles.

Table 7. Parameters list with 80-particle size.

Parameters	Value
Particle size	80
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
w	0.729
c_1	2
c_2	2

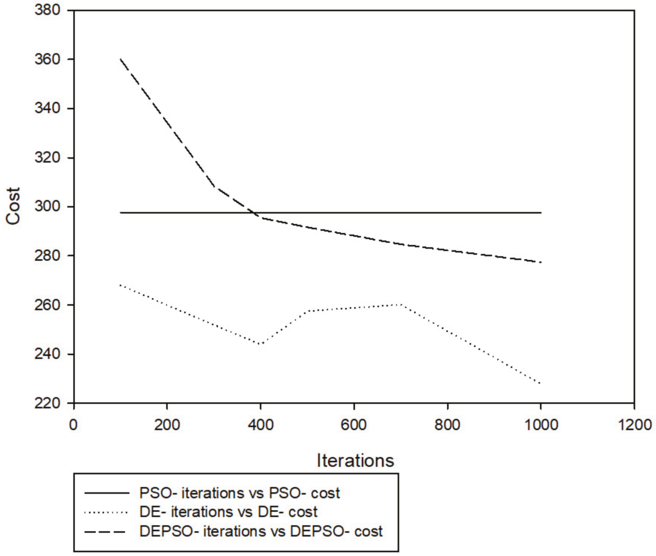


Figure 14. Cost performance of DE, DEPSO, and PSO algorithms at 80 particles.

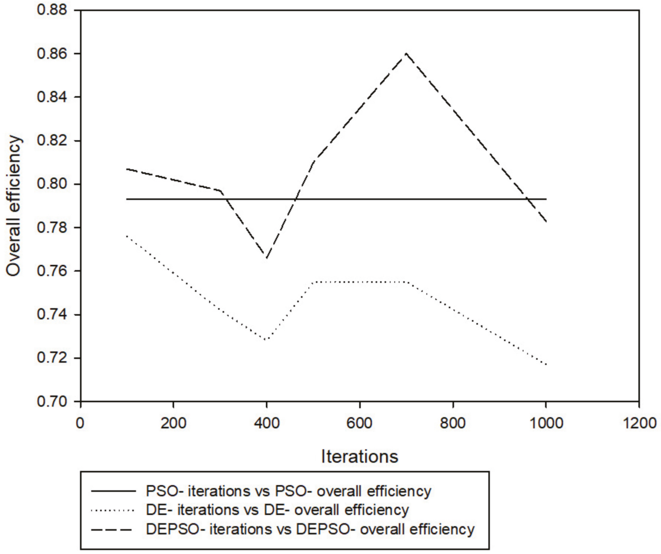


Figure 15. Overall efficiency of DE, DEPSO, and PSO algorithms at 80 particles.

Table 8. Parameters list with 90-particle size.

Parameters	Value
Particle size	90
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
w	0.729
c_1	2
c_2	2

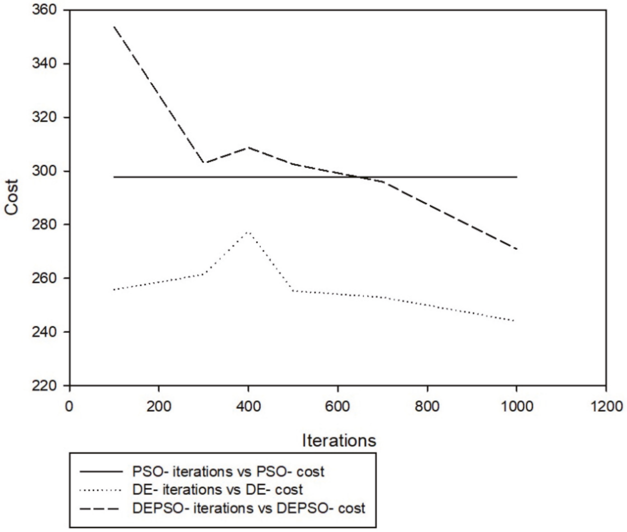


Figure 16. Cost performance of DE, DEPSO, and PSO algorithms at 90 particles.

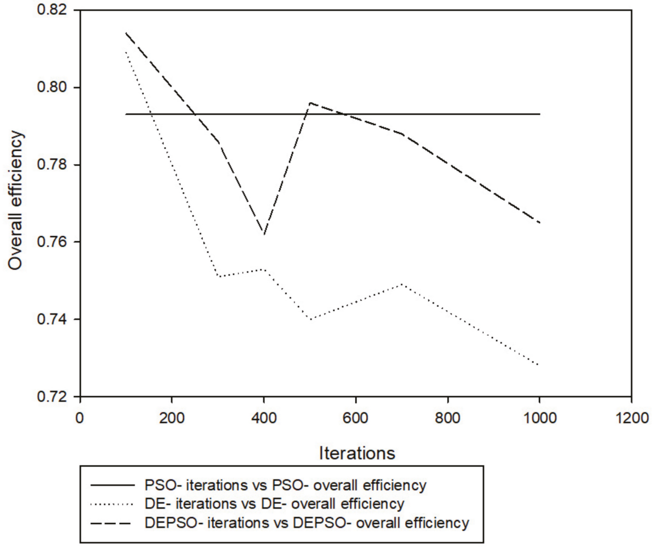


Figure 17. Overall efficiency of DE, DEPSO, and PSO algorithms at 90 particles.

Table 9. Parameters list with 100-particle size.

Parameters	Value
Particle size	100
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
w	0.729
c_1	2
c_2	2

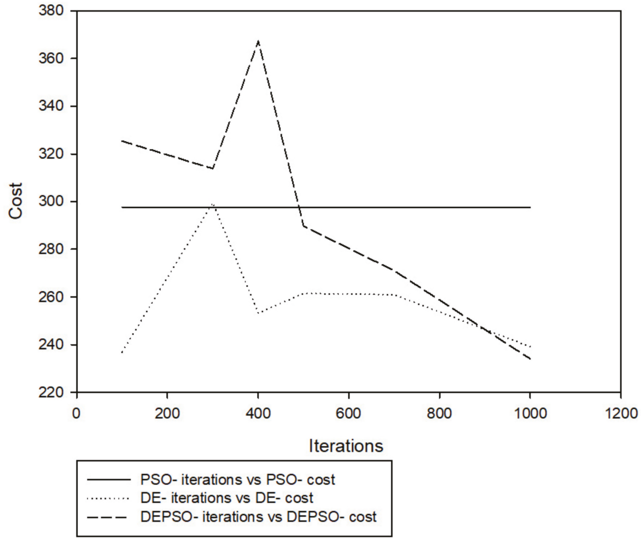


Figure 18. Cost performance of DE, DEPSO, and PSO algorithms at 100 particles.

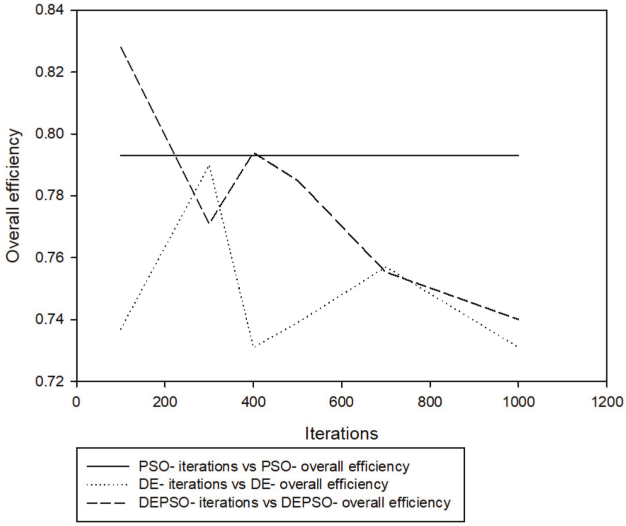


Figure 19. Overall efficiency of DE, DEPSO, and PSO algorithms at 100 particles.

Table 10. Lowest cost value.

	PSO		**DE**		**DEPSO**	
	Cost	**Overall Efficiency**	**Cost**	**Overall Efficiency**	**Cost**	**Overall Efficiency**
	297.715	0.793	218.938	0.716	230	0.70
Iteration Number	100–1000		1000		1000	
Particle Size	50–100		70		40	

The optimized cyclone geometry, using PSO, DE, and hybrid DEPSO algorithms, is shown in Tables 11–14 below.

Table 11. PSO parameters used for design parameter optimization.

Parameters	**Value**
Maximum iteration	1000
Particle size	50
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
w	0.729
c_1	2
c_2	2

Table 12. DE parameters used for design parameter optimization.

Parameters	Value
Maximum iteration	1000
Particle size	70
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
w	0.729
c_1	2
c_2	2

Table 13. DEPSO Parameters used for design parameter optimization.

Parameters	Value
Maximum iteration	1000
Particle size	40
Dimensions	7
Probability of crossover range, C_r	0.5
Scaling factor	1.0
w	0.729
c_1	2
c_2	2

Table 14. Design parameters.

Design Parameters are in the Ratio of D_c	Analytical Method (Stairmand Model)	PSO	DE	Hybrid DEPSO
Overall cyclone height	4.0	3.925	3.371	3.884
Inlet height, a_o	0.5	0.491	0.494	0.489
Gas outlet diameter, D_{eo}	0.5	0.168	0.411	0.309
Cylindrical height of cyclone, h_o	1.5	0.977	0.817	1.006
Inlet width, b_o	0.2	0.213	0.242	0.209
Gas outlet length, S_o	0.5	0.674	1.177	1.171
Dust outlet diameter, B_o	0.375	0.485	0.447	0.179
Pressure drop (Pa)	1000	469.034	364.668	493.24
Inlet velocity m/s	19.86	11.066	9.706	11.36
Cut size d_{pc} (µm)	8.67855×10^{-6}	1.021×10^{-5}	1.26×10^{-5}	9.96×10^{-6}
Cost/s (Naira/s)	-	297.715	218.938	230
Overall efficiency	0.890	0.793	0.716	0.70
Particle size	-	50	70	40

The following can be observed in Table 14 above:

1. Due to the problem of being a minimization type, each algorithm's efficiency is lower compared to the Stairmand's model. The pressure drop of DE is the lowest, followed by PSO and DEPSO. Moreover, each algorithm's pressure drop is considerably lower than the Stairmand's model. An increase in pressure drop will lead to an increase in separation efficiency, higher capacity, and cleaner overflow.
2. Another factor that leads to lower efficiency is the inlet velocity of the cyclone; as seen in Table 14 above, all algorithms have a lower inlet velocity when compared to the Stairmand's model. An increase in inlet velocity leads to an increase in separation efficiency, thereby leading to an increase in the overall efficiency, because of higher resultant centrifugal force.

7. Conclusions

This research utilized a hybrid particle swarm optimization approach to design a gas cyclone with design parameters at a low cost. When compared to PSO individually, as shown in Table 14, the adoption of a hybridised particle swarm optimization technique produced a lower cost of 230 naira/s, while the use of DE had a cost value of 218.938 naira/s.

In this research, it can be concluded that:

- The PSO algorithm, when applied to complex optimization problems, can be improved via the combination with other algorithms, such as the DE algorithm, thereby improving the distribution of information between candidates, due to the crossover feature present within it.
- When both the cognitive component c_1 and the social component c_2 have the same value, both exploration and exploitation are evenly balanced. Furthermore, the inertia weight, w, when less than one, i.e., w < 1, has an effect on the distribution of information across the particles. In addition, when inertia weight is less than one, i.e., w < 1, the velocity of the particle decreases until it reaches zero, thereby causing exploitation to gradually increase and exploration to gradually decrease. In most cases, changing the value of inertia weight, w, during optimization will balance out exploration and exploitation. Some studies, such as [24–26], have developed an inertia weight application.
- When compared to PSO, DEPSO had a better cost value but lower efficiency.
- As shown in Table 14, DE and DEPSO algorithms produced suitable results for design purposes with a low cost, while PSO showed the highest cost value.
- Due to the problem set being of a minimization type, the efficiency suffers considerably, but if the problem was of a maximization type, of which maximum overall efficiency is required, the cost and overall efficiency would be considerably high, provided that the cost of operation is not the primary focus.
- Having made use of the DE/rand/1 strategy with PSO to optimize the cost of operation of the gas cyclone, various other strategies of DE, such as DE/best/1, DE/rand/2, DE/best/2, and DE/target-to-best/1, can also be used.
- PSO is a concept which can be used to solve numerous optimization problems and can be improved through combination with other algorithms. Algorithms such as GA, ACO, TLBO, and so forth can, in fact, be explored to form a hybridised PSO algorithm. Some research on such combinations includes [27–33].

Author Contributions: Conceptualization, X.S. and D.C.I.; methodology, D.C.I.; writing—original draft preparation, D.C.I.; writing-review and editing, X.S. and D.C.I.; supervision, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

ACO	Ant Colony Optimization
DE	Differential Evolution
DEPSO	Differential Evolution PSO
EC	Evolutionary Computation
EA	Evolutionary Algorithm
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
TLBO	Teaching-Learning Based Optimization

References

1. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE international Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
2. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
3. Liu, J.; Lampinen, J. On setting the control parameter of the differential evolution method. In Proceedings of the 8th International Conference on Soft Computing (MENDEL'02), Brno, Czech Republic, 8–10 June 2002; pp. 11–18.
4. Tang, Y.; Gao, H.; Zou, W.; Kurths, J. Identifying controlling nodes in neuronal networks in different scales. *PLoS ONE* **2012**, *7*, e41375. [CrossRef]
5. Mallipeddi, R.; Suganthan, P.; Pan, Q.-K.; Tasgetiren, M.F. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl. Soft Comput.* **2011**, *11*, 1679–1696. [CrossRef]
6. Das, S.; Konar, A. Automatic image pixel clustering with an improved differential evolution. *Appl. Soft Comput.* **2009**, *9*, 226–236. [CrossRef]
7. Storn, R. Differential evolution design of an IIR-filter. In Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), Nagoya, Japan, 20–22 May 1996; pp. 268–273.
8. Varadarajan, M.; Swarup, K. Differential evolution approach for optimal reactive power dispatch. *Appl. Soft Comput.* **2008**, *8*, 1549–1561. [CrossRef]
9. Mallipeddi, R.; Suganthan, P.N. Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies. In Proceedings of the Swarm Evolutionary and Memetic Computing Conference, Chennai, India, 16–18 December 2010; pp. 71–78.
10. Brest, J.; Greiner, S.; Bôskovi'c, B.; Mernik, M.; Zumer, V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **2006**, *10*, 646–657. [CrossRef]
11. Yang, Z.; Tang, K.; Yao, X. Self-adaptive differential evolution with neighborhood search. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC'08), Hong Kong, China, 1–6 June 2008; pp. 1110–1116.
12. Zhang, J.; Sanderson, A.C. JADE: Adaptive Differential Evolution with Optional External Archive. In *IEEE Transactions on Evolutionary Computation*; IEEE: Manhattan, NY, USA, 2009; Volume 13, pp. 945–958.
13. Pant, M.; Thangaraj, R.; Abraham, A. DE-PSO: A new hybrid meta-heuristic for solving global optimization problems. *New Math. Nat. Comput.* **2011**, *7*, 363–381. [CrossRef]
14. Kannan, S.; Slochanal, S.R.; Subbaraj, P.; Padhy, N.P. Application of particle swarm optimization technique and its variants to generation expansion planning problem. *Electr. Power Syst. Res.* **2004**, *70*, 203–210. [CrossRef]
15. Talbi, H.; Batouche, M. Hybrid particle swarm with differential evolution for multimodal image registration. In Proceedings of the IEEE International Conference on Industrial Technology (ICIT '04), Hammamet, Tunisia, 1 December 2004; pp. 1567–1572.
16. Omran, M.G.H.; Engelbrecht, A.P.; Salman, A. Differential evolution based particle swarm optimization. In Proceedings of the IEEE Swarm Intelligence Symposium (SIS '07), Honolulu, HI, USA, 1–5 April 2007; pp. 112–119.
17. Hao, Z.F.; Guo, G.H.; Huang, H. A particle swarm optimization algorithm with differential evolution. In Proceedings of the 6th International Conference on Machine Learning and Cybernetics (ICMLC '07), Hong Kong, China, 19–22 August 2007; pp. 1031–1035.
18. Wang, L. Theoretical Study of Cyclone Design. Ph.D. Thesis, Texas A&M University, College Station, TX, USA, 2004. Available online: https://core.ac.uk/download/pdf/147123938.pdf (accessed on 2 October 2021).
19. Mao, B.; Xie, Z.; Wang, Y.; Handroos, H.; Wu, H. A Hybrid Strategy of Differential Evolution and Modified Particle Swarm Optimization for Numerical Solution of a Parallel Manipulator. *Math. Probl. Eng.* **2018**, *2018*, 1–9. [CrossRef]
20. Wang, L.; Parnell, C.B.; Shaw, B.W. 1D2D, 1D3D, 2D2D cyclone fractional efficiency curves for fine dust. In Proceedings of the 2000 Beltwide Cotton Conferences, San Antonio, TX, USA, 4−8 January 2000.
21. Wang, L.; Parnell, C.B.; Shaw, B.W. *A New Theoretical Approach for Predicting Number of Turns and Cyclone Pressure Drop*; ASAE: Washington, DC, USA; St. Joseph, MI, USA, 2001.
22. Wang, L.; Parnell, C.B.; Oemler, J.A.; Shaw, B.W.; Lacey, R.E. *Analysis of Cyclone Collection Efficiency*; ASAE: Washington, DC, USA; St. Joseph, MI, USA, 2003.
23. Lapple, C.E. Processes use many collector types. *Chem. Eng.* **1951**, *58*, 144–151.
24. Müller, P.K.S.; Airaghi, S.; Marchetto, J. Optimization algorithms based on a model of bacterial chemotaxis. In *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, 1st ed.; A Bradford Book: London, UK, 2000; pp. 375–384. Available online: https://direct.mit.edu/books/book/4493/From-Animals-to-Animats-6Proceedings-of-the-Sixth (accessed on 2 October 2021).
25. Passino, K.M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control. Syst. Mag.* **2002**, *22*, 52–67.
26. Karaboga, D.; Basturk, B. Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems. In *Foundations of Fuzzy Logic and Soft Computing*; Melin, P., Castillo, O., Aguilar, L.T., Pedrycz, W., Eds.; Springer: New York, NY, USA, 2007; pp. 789–798.
27. Mesloub, S.; Mansour, A. Hybrid PSO and GA for global maximization. *Int. J. Open Probl. Comput. Sci. Math.* **2009**, *2*, 597–608.
28. Premalatha, K.; Natarajan, A.M. Hybrid PSO and GA models for document clustering. *Int. J. Adv. Soft Comput. Its Appl.* **2010**, *2*, 302–320.

29. Wang, H.; Li, Y. Hybrid teaching-learning-based PSO for trajectory optimisation. *Electron. Lett.* **2017**, *53*, 777–779. [CrossRef]
30. Kao, Y.; Chen, M.-H.; Huang, Y.-T. A Hybrid Algorithm Based on ACO and PSO for Capacitated Vehicle Routing Problems. *Math. Probl. Eng.* **2012**, *2012*, 1–17. [CrossRef]
31. Dai, Q. A hybrid PSO/ACO algorithm for land cover classification. In Proceedings of the 2nd International Conference on Information Science and Engineering, Hangzhou, China, 3–5 December 2010; pp. 3428–3430.
32. Menghour, K.; Souici-Meslati, L. Hybrid ACO-PSO Based Approaches for Feature Selection. *Int. J. Intell. Eng. Syst.* **2016**, *9*, 65–79. [CrossRef]
33. Azad-Farsani, E.; Zare, M.; Azizipanah-Abarghooee, R.; Askarian-Abyaneh, H. A new hybrid CPSO-TLBO optimization algorithm for distribution network reconfiguration. *J. Intell. Fuzzy Syst.* **2014**, *26*, 2175–2184. [CrossRef]

symmetry

MDPI

Article

Practical Criteria for $\mathcal{H}$-Tensors and Their Application

Min Li, Haifeng Sang *, Panpan Liu and Guorui Huang

School of Mathematics and Statistics, Beihua University, Jilin 132013, China; limin@beihua.edu.cn (M.L.); liupp@beihua.edu.cn (P.L.); shf011@163.com (G.H.)
* Correspondence: sanghaifeng@beihua.edu.cn

Abstract: Identifying the positive definiteness of even-order real symmetric tensors is an important component in tensor analysis. $\mathcal{H}$-tensors have been utilized in identifying the positive definiteness of this kind of tensor. Some new practical criteria for identifying $\mathcal{H}$-tensors are given in the literature. As an application, several sufficient conditions of the positive definiteness for an even-order real symmetric tensor were obtained. Numerical examples are given to illustrate the effectiveness of the proposed method.

Keywords: $\mathcal{H}$-tensors; real symmetric tensors; positive definiteness; irreducible

1. Introduction

Tensor theory is widely used in digital signal processing, medical image processing, data mining, quantum entanglement, and other fields [1–7]. $\mathcal{H}$-tensor theory is an integral part of tensor theory. It plays an important role in physics such as control theory and dynamic control systems and in mathematics such as the numerical solution of partial differential equations and the degree of discretization of nonlinear parabolic equations [8–13].

Let $\mathbb{C}(\mathbb{R})$ be the complex (real) field and $\mathbb{N} = \{1, 2, \ldots, n\}$. A complex (real)-order m dimension n tensor $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ consists of n^m complex (real) entries:

$$a_{i_1 i_2 \ldots i_m} \in \mathbb{C}(\mathbb{R}),$$

where $i_j = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. A tensor $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ is called symmetric [14], if:

$$a_{i_1 i_2 \ldots i_m} = a_{\pi(i_1 i_2 \ldots i_m)}, \forall \pi \in \Pi_m,$$

where Π_m is the permutation group of m indices. Furthermore, a tensor $\mathcal{I} = (\delta_{i_1 i_2 \ldots i_m})$ is called the unit tensor [15], if its entries:

$$\delta_{i_1 i_2 \ldots i_m} = \begin{cases} 1, \ if \ \ i_1 = i_2 = \ldots = i_m, \\ 0, \ \ otherwise. \end{cases}$$

Let $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ be a tensor with order m and dimension n. If there exist a complex number λ and a non-zero complex vector $x = (x_1, x_2, \ldots, x_n)^T$ that are solutions of the following homogeneous polynomial equations:

$$\mathcal{A}x^{m-1} = \lambda x^{[m-1]},$$

then we call λ an eigenvalue of $\mathcal{A}$ and x an eigenvector of $\mathcal{A}$ associated with λ [14,16–20] , and $\mathcal{A}x^{m-1}$ and $\lambda x^{[m-1]}$ are vectors, whose ith components are:

$$(\mathcal{A}x^{m-1})_i = \sum_{i_2, i_3, \ldots, i_m \in \mathbb{N}} a_{i i_2 i_3 \cdots i_m} x_{i_2} x_{i_3} \cdots x_{i_m}$$

Citation: Li, M.; Sang, H.; Liu, P.; Huang, G. Practical Criteria for $\mathcal{H}$-Tensors and Their Application. *Symmetry* **2022**, *14*, 155. https://doi.org/10.3390/sym14010155

Academic Editor: Roman Orus

Received: 15 December 2021
Accepted: 11 January 2022
Published: 13 January 2022

and:

$$(x^{[m-1]})_i = x_i^{m-1}.$$

In particular, if λ and x are restricted to the real field, then we call λ an H-eigenvalue of $\mathcal{A}$ and x an H-eigenvector of $\mathcal{A}$ associated with λ [14].

It is known that an mth-degree homogeneous polynomial of n variables $f(x)$ can be denoted as:

$$f(x) = \sum_{i_1,i_2,\ldots,i_m \in \mathbb{N}} a_{i_1 i_2 \cdots i_m} x_{i_1} x_{i_2} \cdots x_{i_m},$$

where $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$. The homogeneous polynomial $f(x)$ can be expressed as the tensor product of a symmetric tensor $\mathcal{A}$ with order m and dimension n and x^m defined by:

$$f(x) \equiv \mathcal{A}x^m = \sum_{i_1,i_2,\ldots,i_m \in \mathbb{N}} a_{i_1 i_2 \cdots i_m} x_{i_1} x_{i_2} \cdots x_{i_m},$$

where $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$. When m is even, $f(x)$ is called positive definite if $f(x) > 0$, for any $x \in \mathbb{R}^n \setminus \{0\}$. The symmetric tensor $\mathcal{A}$ is called positive definite if $f(x)$ is positive definite [5].

It is well known that the positive definiteness of a multivariate polynomial $f(x)$ plays an important role in the stability study of non-linear autonomous systems [12,21]. However, for $n > 3$ and $m > 4$, it is a hard problem to identify the positive definiteness of such a multivariate form. To solve this problem, Qi [14] pointed out that $f(x) \equiv \mathcal{A}x^m$ is positive definite if and only if the real symmetric tensor $\mathcal{A}$ is positive definite and provided an eigenvalue method to verify the positive definiteness of $\mathcal{A}$ when m is even ([14], Theorem 1.1).

Let $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ be a complex tensor with order m and dimension n; we denote:

$$R_i(A) = \sum_{\substack{i_2,\ldots,i_m \in \mathbb{N} \\ \delta_{i i_2 \cdots i_m} = 0}} |a_{i i_2 \cdots i_m}| = \sum_{i_2,\ldots,i_m \in \mathbb{N}} |a_{i i_2 \cdots i_m}| - |a_{i i \cdots i}|, \quad \forall i \in \mathbb{N}.$$

Definition 1 ([14])**.** *Let $\mathcal{A}$ be a tensor with order m and dimension n. $\mathcal{A}$ is called a diagonally dominant tensor if $|a_{ii\cdots i}| \geq R_i(\mathcal{A}), \quad \forall i \in \mathbb{N}$. $\mathcal{A}$ is called a strictly diagonally dominant tensor if $|a_{ii\cdots i}| > R_i(\mathcal{A}), \quad \forall i \in \mathbb{N}$.*

Definition 2 ([22])**.** *Let $\mathcal{A}$ be a complex tensor with order m and dimension n. $\mathcal{A}$ is called an $\mathcal{H}$-tensor if there is a positive vector $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$, such that:*

$$|a_{ii\cdots i}| x_i^{m-1} > \sum_{\substack{i_2,\ldots,i_m \in \mathbb{N} \\ \delta_{i i_2 \cdots i_m} = 0}} |a_{i i_2 \cdots i_m}| x_{i_2} \cdots x_{i_m}, \quad \forall i \in \mathbb{N}.$$

Definition 3 ([23])**.** *Let $\mathcal{A}$ be a complex tensor with order m and dimension n, $X = \operatorname{diag}(x_1, x_2, \ldots, x_n)$. Denote $\mathcal{B} = (b_{i_1 i_2 \cdots i_m}) = \mathcal{A}X^{m-1}$,*

$$b_{i_1 i_2 \cdots i_m} = a_{i_1 i_2 \cdots i_m} x_{i_2} \cdots x_{i_m}, \quad \forall i_j \in \mathbb{N}, \quad j = 1, 2, \ldots, m.$$

We call $\mathcal{B}$ the product of the tensor $\mathcal{A}$ and the matrix X.

Lemma 1 ([14])**.** *Let $\mathcal{A}$ be an even-order real symmetric tensor, then $\mathcal{A}$ is positive definite if and only if all of its H-eigenvalues are positive.*

From Lemma 1, we can verify the positive definiteness of an even-order symmetric tensor $\mathcal{A}$ (the positive definiteness of the mth-degree homogeneous polynomial $f(x)$) by computing the H-eigenvalues of $\mathcal{A}$. In [6,24,25], for a non-negative tensor, some algorithms were provided to compute its largest eigenvalue. In [1,26], based on semi-definite programming approximation schemes, some algorithms were also given to compute the eigenvalues for general tensors with moderate sizes. However, it is not easy to compute all these H-eigenvalues when m and n are very large. Recently, by introducing the definition of $\mathcal{H}$-tensor, References [22,27] and Li et al. [27] provided a practical sufficient condition for identifying the positive definiteness of an even-order symmetric tensor (see Lemmas 2, 4, and 5).

Lemma 2 ([22])**.** *If $\mathcal{A} = (a_{i_1 i_2 \cdots i_m})$ is a strictly diagonally dominant tensor, then $\mathcal{A}$ is an $\mathcal{H}$-tensor.*

Lemma 3 ([27])**.** *Let $\mathcal{A} = (a_{i_1 i_2 \cdots i_m})$ be an even-order real symmetric tensor of order m and dimension n with $a_{ii\cdots i} > 0$ for all $i \in \mathbb{N}$. If $\mathcal{A}$ is an $\mathcal{H}$-tensor, then $\mathcal{A}$ is positive definite.*

Lemma 4 ([27])**.** *Let $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ be a complex tensor with order m and dimension n. If there exists a positive diagonal matrix X such that $\mathcal{A}X^{m-1}$ is an $\mathcal{H}$-tensor, then $\mathcal{A}$ is an $\mathcal{H}$-tensor.*

Lemma 5. *Let $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ be a complex tensor with order m and dimension n. If $\mathcal{A}$ is an $\mathcal{H}$-tensor, then there exists at least one index $i_0 \in \mathbb{N}$ such that $|a_{i_0 i_0 \cdots i_0}| > R_{i_0}(\mathcal{A})$.*

Proof of Lemma 5. According to Definition 2, there is a positive vector $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$, such that:

$$|a_{ii\cdots i}|x_i^{m-1} > \sum_{\substack{i_2,\ldots,i_m \in \mathbb{N} \\ \delta_{ii_2\cdots i_m}=0}} |a_{ii_2\cdots i_m}|x_{i_2}\cdots x_{i_m}, \quad \forall i \in \mathbb{N}.$$

Denote $x_{i_0} = \min\limits_{i\in\mathbb{N}}\{x_i\}$, then for the index $i_0 \in \mathbb{N}$, we have:

$$|a_{i_0 i_0\cdots i_0}| > \sum_{\substack{i_2,\ldots,i_m \in \mathbb{N} \\ \delta_{ii_2\cdots i_m}=0}} |a_{i_0 i_2\cdots i_m}|\frac{x_{i_2}}{x_{i_0}}\cdots\frac{x_{i_m}}{x_{i_0}} \geq \sum_{\substack{i_2,\ldots,i_m \in \mathbb{N} \\ \delta_{ii_2\cdots i_m}=0}} |a_{i_0 i_2\cdots i_m}| = R_{i_0}(\mathcal{A}).$$

The proof is complete. □

2. Practical Criteria for the $\mathcal{H}$-Tensor

Throughout this paper, we use the following definitions and notation.

$$\mathbb{N} = \{1,2,\cdots,n\} = \bigcup_{i=1}^{k} \mathbb{N}_i, \mathbb{N}_i \cap \mathbb{N}_j = \emptyset, 1 \leq i \neq j \leq \ k.$$
$$\forall t \in \{1,2,\cdots,k\}, \mathbb{N}_t^{m-1} = \{i_2 i_3 \cdots i_m | i_j \in \mathbb{N}_t, j = 1,2,\cdots,m\}.$$
$$\mathbb{N}^{m-1} \backslash \mathbb{N}_t^{m-1} = \{i_2 i_3 \cdots i_m | i_2 i_3 \cdots i_m \in \mathbb{N}^{m-1}, i_2 i_3 \cdots i_m \notin \mathbb{N}_t^{m-1}\}.$$

For all $i \in \mathbb{N}$, there exists a constant $t \in \{1,2,\cdots,k\}$ that satisfies $i \in \mathbb{N}_t$. Denote:

$$\alpha_{\mathbb{N}_t}^{(i)} = \sum_{\substack{i_2\cdots i_m \in \mathbb{N}_t^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |\ a_{ii_2\cdots i_m}\ |,\ \overline{\alpha}_{\mathbb{N}_t}^{(i)} = \sum_{i_2,\cdots,i_m \in \mathbb{N}^{m-1} \backslash \mathbb{N}_t^{m-1}} |\ a_{ii_2\cdots i_m}\ | = R_i(\mathcal{A}) - \alpha_{\mathbb{N}_t}^{(i)}.$$

$$\mathbb{N}^* = \{i|\ |\ a_{ii\cdots i}\ | > R_i(\mathcal{A}), i \in \mathbb{N}\},$$

$$\mathbb{N}^+ = \{i|\ |\ a_{ii\cdots i}\ | > \alpha_{\mathbb{N}_t}^{(i)}, i \in \mathbb{N}_t \subseteq \mathbb{N}\}, \mathbb{N}^0 = \{i|\ |\ a_{ii\cdots i}\ | = \alpha_{\mathbb{N}_t}^{(i)}, i \in \mathbb{N}_t \subseteq \mathbb{N}\},$$

$$\mathbb{J}^+ = \{(i,j)|(|\ a_{ii\cdots i}\ | - \alpha_{\mathbb{N}_t}^{(i)})(|\ a_{jj\cdots j}\ | - \alpha_{\mathbb{N}_s}^{(j)}) > \overline{\alpha}_{\mathbb{N}_t}^{(i)} \cdot \overline{\alpha}_{\mathbb{N}_s}^{(j)}, i \in \mathbb{N}_t, j \in \mathbb{N}_s, 1 \leq t \neq s \leq k\}$$

$$\mathbb{J}^0 = \{(i,j)|(|\ a_{ii\cdots i}\ | - \alpha_{\mathbb{N}_t}^{(i)})(|\ a_{jj\cdots j}\ | - \alpha_{\mathbb{N}_s}^{(j)}) = \overline{\alpha}_{\mathbb{N}_t}^{(i)} \cdot \overline{\alpha}_{\mathbb{N}_s}^{(j)}, i \in \mathbb{N}_t, j \in \mathbb{N}_s, 1 \leq t \neq s \leq k\}$$

Definition 4. *Let $\mathcal{A} = (a_{i_1 i_2 \cdots i_m})$ be a complex tensor with order m and dimension n. For all $i \in \mathbb{N}_t$, $j \in \mathbb{N}_s, 1 \leq t \neq s \leq k$, if:*

$$(| a_{ii\cdots i} | - \alpha^{(i)}_{\mathbb{N}_t})(| a_{jj\cdots j} | - \alpha^{(j)}_{\mathbb{N}_s}) \geq \overline{\alpha}^{(i)}_{\mathbb{N}_t} \cdot \overline{\alpha}^{(j)}_{\mathbb{N}_s}, \tag{1}$$

we call $\mathcal{A}$ a locally double-diagonally dominant tensor and denote $\mathcal{A} \in LDD_0T$. If:

$$(| a_{ii\cdots i} | - \alpha^{(i)}_{\mathbb{N}_t})(| a_{jj\cdots j} | - \alpha^{(j)}_{\mathbb{N}_s}) > \overline{\alpha}^{(i)}_{\mathbb{N}_t} \cdot \overline{\alpha}^{(j)}_{\mathbb{N}_s}, \tag{2}$$

we call $\mathcal{A}$ a strictly locally double-diagonally dominant tensor and denote $\mathcal{A} \in LDDT$.

Remark 1. *If $\mathcal{A} \in LDD_0T$ and $\mathbb{N}^* \neq \emptyset$, then $\mathbb{N} = \mathbb{N}^0 \cup \mathbb{N}^+$; if $\mathcal{A} \in LDDT$ and $\mathbb{N}^* \neq \emptyset$, then $\mathbb{N} = \mathbb{N}^+$.*

Lemma 6. *Let $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ be a complex tensor with order m and dimension n. If $\mathcal{A} \in LDDT$ and $\mathbb{N}^* \neq \emptyset$, then there exists at most one $l_0 \in \{1, 2, \cdots, k\}$ such that,*

$$| a_{ii\cdots i} | - \alpha^{(i)}_{\mathbb{N}_{l_0}} \leq \overline{\alpha}^{(i)}_{\mathbb{N}_{l_0}}, \ \ \forall i \in \mathbb{N}_{l_0}.$$

Proof of Lemma 6. If there exists $l_0^1 \neq l_0^2 \in \{1, 2, \cdots, k\}$, such that for all $i \in \mathbb{N}_{l_0^1}, j \in \mathbb{N}_{l_0^2}$,

$$| a_{ii\cdots i} | - \alpha^{(i)}_{\mathbb{N}_{l_0^1}} \leq \overline{\alpha}^{(i)}_{\mathbb{N}_{l_0^1}}, \quad | a_{jj\cdots j} | - \alpha^{(j)}_{\mathbb{N}_{l_0^2}} \leq \overline{\alpha}^{(j)}_{\mathbb{N}_{l_0^2}}.$$

Notice that $\mathcal{A} \in LDDT$ and $\mathbb{N}^* \neq \emptyset$. By Remark 1, we have:

$$| a_{ii\cdots i} | - \alpha^{(i)}_{\mathbb{N}_{l_0^1}} > 0, \quad | a_{jj\cdots j} | - \alpha^{(j)}_{\mathbb{N}_{l_0^2}} > 0.$$

These imply:

$$(| a_{ii\cdots i} | - \alpha^{(i)}_{\mathbb{N}_{l_0^1}}) \cdot (| a_{jj\cdots j} | - \alpha^{(j)}_{\mathbb{N}_{l_0^2}}) \leq \overline{\alpha}^{(i)}_{\mathbb{N}_{l_0^1}} \cdot \overline{\alpha}^{(j)}_{\mathbb{N}_{l_0^2}},$$

which contradicts $\mathcal{A} \in LDDT$. The proof is complete. □

Remark 2. *Based on Lemma 6, when $\mathcal{A} \in LDDT$ and $\mathbb{N}^* \neq \emptyset$, we always assume that $\mathbb{N}_1 = \{i | \, | a_{ii\cdots i} | - \alpha^{(i)}_{\mathbb{N}_t} \leq \overline{\alpha}^{(i)}_{\mathbb{N}_t}, \ i \in \mathbb{N}\}$.*

Lemma 7. *Let $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ be a complex tensor with order m and dimension n, $\mathcal{A} \in LDDT$ and $\mathbb{N}^* \neq \emptyset$. If $\mathbb{N}_1 = \emptyset$, then $\mathcal{A}$ is an $\mathcal{H}$-tensor.*

Proof of Lemma 7. Since $\mathcal{A} \in LDDT, \mathbb{N}^* \neq \emptyset$ and $\mathbb{N}_1 = \emptyset$. For all $i \in \mathbb{N} = \mathbb{N}_2 \cup \mathbb{N}_3 \cup \cdots \cup \mathbb{N}_k$, we have:

$$| a_{ii\cdots i} | - \alpha^{(i)}_{\mathbb{N}_t} > \overline{\alpha}^{(i)}_{\mathbb{N}_t}, t \in \{2, 3, \cdots, k\}.$$

This implies:

$$| a_{ii\cdots i} | > \alpha^{(i)}_{\mathbb{N}_t} + \overline{\alpha}^{(i)}_{\mathbb{N}_t} = R_i(\mathcal{A}), i \in \mathbb{N}.$$

By Lemma 2, $\mathcal{A}$ is an $\mathcal{H}$-tensor. □

Theorem 1. *Let $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ be a complex tensor with order m and dimension n, $\mathcal{A} \in LDDT$ and $\mathbb{N}^* \neq \emptyset$. If for all $i \in \mathbb{N}_t, t \in \{2, 3, \cdots, k\}$, $\ j \in \mathbb{N}_1 \neq \emptyset$, we have:*

$$\begin{aligned}&(| a_{ii\cdots i} | -\alpha^{(i)}_{\mathbb{N}_t})(| a_{jj\cdots j} | - \sum_{\substack{i_2i_3\cdots i_m\in\mathbb{N}^{m-1}\backslash(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}\\ \delta_{ji_2\cdots i_m}=0}} | a_{ji_2\cdots i_m} |)\\ &> \overline{\alpha}^{(i)}_{\mathbb{N}_t}\cdot(\sum_{i_2i_3\cdots i_m\in(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}} | a_{ji_2i_3\cdots i_m} |),\end{aligned} \tag{3}$$

then $\mathcal{A}$ is an $\mathcal{H}$-tensor.

Proof of Theorem 1. From the inequality (3), there exists a positive constant d such that:

$$\min_{\substack{i\in\mathbb{N}_t\\ t\in\{2,3,\cdots,k\}}}\{\frac{| a_{ii\cdots i} | -\alpha^{(i)}_{\mathbb{N}_t}}{\overline{\alpha}^{(i)}_{\mathbb{N}_t}}\} > d > \max_{j\in\mathbb{N}_1}\{\frac{\sum\limits_{i_2i_3\cdots i_m\in(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}} | a_{ji_2i_3\cdots i_m} |}{| a_{jj\cdots j} | - \sum\limits_{\substack{i_2i_3\cdots i_m\in\mathbb{N}^{m-1}\backslash(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}\\ \delta_{ji_2\cdots i_m}=0}} | a_{ji_2\cdots i_m} |}\}. \tag{4}$$

If $\overline{\alpha}^{(i)}_{\mathbb{N}_t} = 0$, we denote $\frac{|a_{ii\cdots i}|-\alpha^{(i)}_{\mathbb{N}_t}}{\overline{\alpha}^{(i)}_{\mathbb{N}_t}} = +\infty$. It is obvious that $d > 1$. Construct a positive diagonal matrix $X = \mathrm{diag}(x_1, x_2, \cdots, x_n)$, where:

$$x_i = \begin{cases}1, \ i\in\mathbb{N}\backslash\mathbb{N}_1,\\ d^{\frac{1}{m-1}}, \ i\in\mathbb{N}_1.\end{cases}$$

Let $\mathcal{B} = \mathcal{A}X^{m-1} = (b_{i_1i_2\cdots i_m})$. For $i \in \mathbb{N}\backslash\mathbb{N}_1 = \{\mathbb{N}_2, \mathbb{N}_3, \cdots, \mathbb{N}_k\}$, by the first inequality in (4),

$$\begin{aligned}| b_{ii\cdots i} | &=| a_{ii\cdots i} | \cdot\overbrace{1\cdot 1\cdots 1}^{m-1}\\ &=| a_{ii\cdots i} |\\ &> \alpha^{(i)}_{\mathbb{N}_t} + \overline{\alpha}^{(i)}_{\mathbb{N}_t}\cdot\ d\\ &= \sum_{\substack{i_2\cdots i_m\in\mathbb{N}_t^{m-1}\\ \delta_{ii_2\cdots i_m}=0}} | a_{ii_2\cdots i_m} | + \sum_{i_2\cdots i_m\in\mathbb{N}^{m-1}\backslash\mathbb{N}_t^{m-1}} | a_{ii_2\cdots i_m} | \overbrace{d^{\frac{1}{m-1}}\cdots d^{\frac{1}{m-1}}}^{m-1}\\ &\geq \sum_{\substack{i_2\cdots i_m\in\mathbb{N}_t^{m-1}\\ \delta_{ii_2\cdots i_m}=0}} | a_{ii_2\cdots i_m} | + \sum_{i_2\cdots i_m\in\mathbb{N}^{m-1}\backslash\mathbb{N}_t^{m-1}} | a_{ii_2\cdots i_m} | x_{i_2}\cdots x_{i_m}\\ &= \sum_{\substack{i_2\cdots i_m\in\mathbb{N}_t^{m-1}\\ \delta_{ii_2\cdots i_m}=0}} | b_{ii_2\cdots i_m} | + \sum_{i_2\cdots i_m\in\mathbb{N}^{m-1}\backslash\mathbb{N}_t^{m-1}} | b_{ii_2\cdots i_m} |\\ &= R_i(\mathcal{B}).\end{aligned}$$

For $i \in \mathbb{N}_1$, by the second inequality in (4),

$$\begin{aligned}
\mid b_{ii\cdots i}\mid &=\mid a_{ii\cdots i}\mid \cdot d \\
&=\mid a_{ii\cdots i}\mid \cdot \overbrace{d^{\frac{1}{m-1}}\cdots d^{\frac{1}{m-1}}}^{m-1} \\
&> d\cdot \sum_{\substack{i_2\cdots i_m\in\mathbb{N}^{m-1}\backslash(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}\\ \delta_{ii_2\cdots i_m}=0}} \mid a_{ii_2\cdots i_m}\mid + \sum_{i_2\cdots i_m\in(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}} \mid a_{ii_2\cdots i_m}\mid \\
&= d(\sum_{\substack{i_2\cdots i_m\in\mathbb{N}_1^{m-1}\\ \delta_{ii_2\cdots i_m}=0}} \mid a_{ii_2\cdots i_m}\mid + \sum_{i_2\cdots i_m\in[\mathbb{N}^{m-1}\backslash(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}]\backslash\mathbb{N}_1^{m-1}} \mid a_{ii_2\cdots i_m}\mid) + \sum_{i_2\cdots i_m\in(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}} \mid a_{ii_2\cdots i_m}\mid \\
&\geq \sum_{\substack{i_2\cdots i_m\in\mathbb{N}_1^{m-1}\\ \delta_{ii_2\cdots i_m}=0}} \mid a_{ii_2\cdots i_m}\mid \cdot \overbrace{d^{\frac{1}{m-1}}\cdots d^{\frac{1}{m-1}}}^{m-1} \\
&+ \sum_{i_2\cdots i_m\in[\mathbb{N}^{m-1}\backslash(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}]\backslash\mathbb{N}_1^{m-1}} \mid a_{ii_2\cdots i_m}\mid x_{i_2}\cdots x_{i_m} + \sum_{i_2\cdots i_m\in(\mathbb{N}\backslash\mathbb{N}_1)^{m-1}} \mid a_{ii_2\cdots i_m}\mid \\
&= \sum_{\substack{i_2\cdots i_m\in\mathbb{N}_1^{m-1}\\ \delta_{ii_2\cdots i_m}=0}} \mid b_{ii_2\cdots i_m}\mid + \sum_{i_2 i_3\cdots i_m\in\mathbb{N}^{m-1}\backslash\mathbb{N}_1^{m-1}} \mid b_{ii_2\cdots i_m}\mid \\
&= R_i(\mathcal{B}).
\end{aligned}$$

Thus, we have proven that:

$$\mid b_{ii\cdots i}\mid > \sum_{\substack{i_2\cdots i_m\in\mathbb{N}^{m-1}\\ \delta_{ii_2\cdots i_m}=0}} \mid b_{ii_2\cdots i_m}\mid = R_i(\mathcal{B}),\ \ i\in\mathbb{N}.$$

i.e., $\mathcal{B}$ is a strictly diagonally dominant tensor. By Lemmas 2 and 4, $\mathcal{A}$ is an $\mathcal{H}$-tensor. The proof is complete. □

Lemma 8. *Let $\mathcal{A}=(a_{i_1 i_2 \ldots i_m})$ be a complex tensor with order m and dimension n. If $\mathbb{N}^*\neq\emptyset$ and for all $i\in\mathbb{N}_t$, $j\in\mathbb{N}_s$, $1\leq t\neq s\leq k$,*

$$(\mid a_{ii\cdots i}\mid -\alpha^{(i)}_{\mathbb{N}_t})(\mid a_{jj\cdots j}\mid -\alpha^{(j)}_{\mathbb{N}_s})\leq \overline{\alpha}^{(i)}_{\mathbb{N}_t}\cdot\overline{\alpha}^{(j)}_{\mathbb{N}_s}, \tag{5}$$

then there exists only one natural number $k_0\in\{1,2,\cdots,k\}$ such that $\mathbb{N}^*\subseteq\mathbb{N}_{k_0}$.

Proof of Lemma 8. Since $\mathbb{N}^*\neq\emptyset$, then there exists one $k_0\in\{1,2,\cdots,k\}$ such that $\mathbb{N}^*\cap\mathbb{N}_{k_0}\neq\emptyset$. If there exists another $k_1\in\{1,2,\cdots,k\}\setminus\{k_0\}$ such that $\mathbb{N}^*\cap\mathbb{N}_{k_1}\neq\emptyset$, then we have:

$$(\mid a_{ii\cdots i}\mid -\alpha^{(i)}_{\mathbb{N}_{k_0}\cap\mathbb{N}^*})(\mid a_{jj\cdots j}\mid -\alpha^{(j)}_{\mathbb{N}_{k_1}\cap\mathbb{N}^*}) > \overline{\alpha}^{(i)}_{\mathbb{N}_{k_0}\cap\mathbb{N}^*}\cdot\overline{\alpha}^{(j)}_{\mathbb{N}_{k_1}\cap\mathbb{N}^*}$$

which contradicts the inequality (5). Therefore, there exists only one natural number $k_0\in\{1,2,\cdots,k\}$ that satisfies $\mathbb{N}^*\subseteq\mathbb{N}_{k_0}$. □

Remark 3. *Based on Lemma 8, when $\mathcal{A}$ satisfies (5) and $\mathbb{N}^*\neq\emptyset$, then we always assume that $\mathbb{N}^*\subseteq\mathbb{N}_k$ (where $\mathbb{N}=\mathbb{N}_1\cup\mathbb{N}_2\cup\cdots\cup\mathbb{N}_k$).*

Theorem 2. *Let $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ be a complex tensor with order m and dimension n and $\mathbb{N}^* \neq \emptyset$. For all $i \in \mathbb{N}_k,\ j \in \mathbb{N}_s, s \in \{1, 2, \cdots, k-1\}$, if:*

$$\begin{aligned} 0 &< (\mid a_{ii\cdots i} \mid - \sum_{\substack{i_2\cdots i_m \in \mathbb{N}^{m-1} \backslash (\mathbb{N} \backslash \mathbb{N}_k)^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} \mid a_{ii_2\cdots i_m} \mid)(\mid a_{jj\cdots j} \mid - \alpha_{\mathbb{N}_s}^{(j)}) \\ &\leq (\sum_{i_2\cdots i_m \in (\mathbb{N} \backslash \mathbb{N}_k)^{m-1}} \mid a_{ii_2\cdots i_m} \mid) \cdot \overline{\alpha}_{\mathbb{N}_s}^{(j)}, \end{aligned} \tag{6}$$

then $\mathcal{A}$ is not an $\mathcal{H}$-tensor.

Proof of Theorem 2. From the inequality (6), there exists a positive constant d_0 such that:

$$\max_{i \in \mathbb{N}_k} \{ \frac{\mid a_{ii\cdots i} \mid - \sum\limits_{\substack{i_2\cdots i_m \in \mathbb{N}^{m-1} \backslash (\mathbb{N} \backslash \mathbb{N}_k)^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} \mid a_{ii_2\cdots i_m} \mid}{\sum\limits_{i_2\cdots i_m \in (\mathbb{N} \backslash \mathbb{N}_k)^{m-1}} \mid a_{ii_2\cdots i_m} \mid} \} \leq d_0 \leq \min_{j \in \mathbb{N} \backslash \mathbb{N}_k} \{ \frac{\overline{\alpha}_{\mathbb{N}_s}^{(j)}}{\mid a_{jj\cdots j} \mid - \alpha_{\mathbb{N}_s}^{(j)}} \}. \tag{7}$$

If $\mid a_{jj\cdots j} \mid - \alpha_{\mathbb{N}_s}^{(j)} = 0, j \in \mathbb{N} \setminus \mathbb{N}_k$, we denote $\frac{\overline{\alpha}_{\mathbb{N}_s}^{(j)}}{|a_{jj\cdots j}| - \alpha_{\mathbb{N}_s}^{(j)}} = +\infty$. Obviously,

$$\max_{i \in \mathbb{N}_k} \{ \frac{\mid a_{ii\cdots i} \mid - \sum\limits_{\substack{i_2\cdots i_m \in \mathbb{N}^{m-1} \backslash (\mathbb{N} \backslash \mathbb{N}_k)^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} \mid a_{ii_2\cdots i_m} \mid}{\sum\limits_{i_2\cdots i_m \in (\mathbb{N} \backslash \mathbb{N}_k)^{m-1}} \mid a_{ii_2\cdots i_m} \mid} \} \geq \max_{i \in \mathbb{N}^*} \{ \frac{\mid a_{ii\cdots i} \mid - \sum\limits_{\substack{i_2\cdots i_m \in \mathbb{N}^{m-1} \backslash (\mathbb{N} \backslash \mathbb{N}_k)^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} \mid a_{ii_2\cdots i_m} \mid}{\sum\limits_{i_2\cdots i_m \in (\mathbb{N} \backslash \mathbb{N}_k)^{m-1}} \mid a_{ii_2\cdots i_m} \mid} \} > 1,$$

so $d_0 > 1$. Construct a positive diagonal matrix $X = \text{diag}(x_1, x_2, \cdots, x_n)$, where:

$$x_i = \begin{cases} 1, \ \ i \in \mathbb{N}_k, \\ d_0^{\frac{1}{m-1}}, \ \ i \in \mathbb{N} \backslash \mathbb{N}_k. \end{cases}$$

Let $\mathcal{B} = \mathcal{A} X^{m-1} = (b_{i_1 i_2 \cdots i_m})$. For $i \in \mathbb{N}^*$, by the first inequality in (7),

$$\begin{aligned} |b_{ii\cdots i}| &= |a_{ii\cdots i}| \cdot \overbrace{1 \cdot 1 \cdots 1}^{m-1} = |a_{ii\cdots i}| \\ &\leq \sum_{\substack{i_2\cdots i_m \in \mathbb{N}^{m-1} \backslash (\mathbb{N} \backslash \mathbb{N}_k)^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |a_{ii_2\cdots i_m}| + d_0 \cdot \sum_{i_2\cdots i_m \in (\mathbb{N} \backslash \mathbb{N}_k)^{m-1}} |a_{ii_2\cdots i_m}| \\ &\leq \sum_{\substack{i_2\cdots i_m \in \mathbb{N}_k^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |a_{ii_2\cdots i_m}| + \sum_{i_2\cdots i_m \in [\mathbb{N}^{m-1} \backslash (\mathbb{N} \backslash \mathbb{N}_k)^{m-1}] \backslash \mathbb{N}_k^{m-1}} |a_{ii_2\cdots i_m}| x_{i_2} \cdots x_{i_m} \\ &+ \sum_{i_2\cdots i_m \in (\mathbb{N} \backslash \mathbb{N}_k)^{m-1}} |a_{ii_2\cdots i_m}| \cdot \overbrace{d^{\frac{1}{m-1}} \cdots d^{\frac{1}{m-1}}}^{m-1} \\ &= \sum_{\substack{i_2\cdots i_m \in \mathbb{N}_k^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |b_{ii_2\cdots i_m}| + \sum_{i_2\cdots i_m \in \mathbb{N}^{m-1} \backslash \mathbb{N}_k^{m-1}} |b_{ii_2\cdots i_m}| \\ &= R_i(B). \end{aligned}$$

For $i \in \mathbb{N}_k \backslash \mathbb{N}^*$,

$$\begin{aligned}
|b_{ii\cdots i}| &= |a_{ii\cdots i}| \cdot \overbrace{1 \cdot 1 \cdots 1}^{m-1} = |a_{ii\cdots i}| \\
&\le \sum_{\substack{i_2\cdots i_m \in \mathbb{N}_k^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |a_{ii_2\cdots i_m}| + \sum_{i_2\cdots i_m \in \mathbb{N}^{m-1}\backslash \mathbb{N}_k^{m-1}} |a_{ii_2\cdots i_m}| \\
&\le \sum_{\substack{i_2\cdots i_m \in \mathbb{N}_k^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |a_{ii_2\cdots i_m}| + \sum_{i_2\cdots i_m \in \mathbb{N}^{m-1}\backslash (\mathbb{N}^{m-1}\backslash \mathbb{N}_k^{m-1})} |a_{ii_2\cdots i_m}| x_{i_2} \cdots x_{i_m} \\
&+ \sum_{i_2\cdots i_m \in \mathbb{N}^{m-1}\backslash \mathbb{N}_k^{m-1}} |a_{ii_2\cdots i_m}| \overbrace{d_0^{\frac{1}{m-1}} \cdots d_0^{\frac{1}{m-1}}}^{m-1} \\
&= \sum_{\substack{i_2\cdots i_m \in \mathbb{N}_k^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |b_{ii_2\cdots i_m}| + \sum_{i_2\cdots i_m \in \mathbb{N}^{m-1}\backslash \mathbb{N}_k^{m-1}} |b_{ii_2\cdots i_m}| \\
&= R_i(B).
\end{aligned}$$

For $i \in \mathbb{N}\backslash\mathbb{N}_k = \mathbb{N}_1 \bigcup \mathbb{N}_2 \bigcup \cdots \bigcup \mathbb{N}_{k-1}$, by the second inequality in (7),

$$\begin{aligned}
|b_{ii\cdots i}| &= d_0 |a_{ii\cdots i}| = |a_{ii\cdots i}| \overbrace{d_0^{\frac{1}{m-1}} \cdots d_0^{\frac{1}{m-1}}}^{m-1} \\
&\le d_0 \cdot \alpha_{\mathbb{N}_s}^{(i)} + \overline{\alpha}_{\mathbb{N}_s}^{(i)} \\
&= \sum_{\substack{i_2\cdots i_m \in \mathbb{N}_s^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |a_{ii_2\cdots i_m}| \overbrace{d_0^{\frac{1}{m-1}} \cdots d_0^{\frac{1}{m-1}}}^{m-1} + \sum_{i_2\cdots i_m \in \mathbb{N}^{m-1}\backslash \mathbb{N}_s^{m-1}} |a_{ii_2\cdots i_m}| \\
&\le \sum_{\substack{i_2\cdots i_m \in \mathbb{N}_s^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |a_{ii_2\cdots i_m}| \overbrace{d_0^{\frac{1}{m-1}} \cdots d_0^{\frac{1}{m-1}}}^{m-1} + \sum_{i_2\cdots i_m \in \mathbb{N}^{m-1}\backslash \mathbb{N}_s^{m-1}} |a_{ii_2\cdots i_m}| x_{i_2} \cdots x_{i_m} \\
&= \sum_{\substack{i_2\cdots i_m \in \mathbb{N}_s^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} |b_{ii_2\cdots i_m}| + \sum_{i_2\cdots i_m \in \mathbb{N}^{m-1}\backslash \mathbb{N}_s^{m-1}} |b_{ii_2\cdots i_m}| \\
&= R_i(B).
\end{aligned}$$

Thus we have proved that:

$$| b_{ii\cdots i} | \le \sum_{\substack{i_2\cdots i_m \in \mathbb{N}^{m-1} \\ \delta_{ii_2\cdots i_m}=0}} | b_{ii_2\cdots i_m} | = R_i(\mathcal{B}), \ \ i \in \mathbb{N}.$$

i.e., $\mathcal{B}$ is not strictly diagonally dominant for all $i \in \mathbb{N}$. By Lemma 5, $\mathcal{B}$ is not an $\mathcal{H}$-tensor. By Lemma 4, $\mathcal{A}$ is not an $\mathcal{H}$-tensor. The proof is complete. □

3. An Algorithm for Identifying $\mathcal{H}$-Tensors

Based on the results in the above section, an algorithm for identifying $\mathcal{H}$-tensors is put forward in this section:

Remark 4.

(a) *$s =$ the total number of tensors, $k_1 =$ the number of $\mathcal{H}$-tensors, $k_2 =$ the number of tensors which are not $\mathcal{H}$-tensor, and $s - k_1 - k_2 =$ the number of tensors, which are not checkable by using Algorithm 1;*

(b) *The calculations of Algorithm 1 only depend on the elements of the tensor, so Algorithm 1 stops after a finite amount of steps.*

Algorithm 1 An algorithm for identifying $\mathcal{H}$-tensors

```
Step 1. Set $k_1 := 0$, $k_2 := 0$, $k_3 := 0$ and $s := 50$.
Step 2. Given a complex tensor $\mathcal{A} = (a_{i_1 \cdots i_m})$. If $k_3 = s$, then output $k_1$ and $k_2$, and stop. Otherwise,
Step 3. Compute $|a_{i \cdots i}|$ and $R_i(\mathcal{A})$ for all $i \in \mathbb{N}$.
Step 4. If $N^* = N$, then print "$\mathcal{A}$ is a $\mathcal{H}$-tensor", and go to Step 5. Otherwise, go to Step 6.
Step 5. Replace $k_1$ by $k_1 + 1$, and replace $k_3$ by $k_3 + 1$, then go to Step 2.
Step 6. If $N^* = \emptyset$, then print "$\mathcal{A}$ is not a $\mathcal{H}$-tensor", and go to Step 7. Otherwise, go to Step 8.
Step 7. Replace $k_2$ by $k_2 + 1$, and replace $k_3$ by $k_3 + 1$, then go to Step 2.
Step 8. Compute $|a_{i \cdots i}|$, $\alpha^{(i)}_{\mathbb{N}_t}$, $\overline{\alpha}^{(i)}_{\mathbb{N}_t}$, $|a_{j \cdots j}|$, $\alpha^{(j)}_{\mathbb{N}_s}$, $\overline{\alpha}^{(j)}_{\mathbb{N}_s}$ for all $i \in \mathbb{N}_t$, $j \in \mathbb{N}_s, 1 \leq t \neq s \leq k$.
Step 9. If Inequality (3) holds, then print "$\mathcal{A}$ is a $\mathcal{H}$-tensor", and go to Step 5. Otherwise,
Step 10. Compute:
    $$\sum_{\substack{i_2 \cdots i_m \in \mathbb{N}^{m-1} \backslash (\mathbb{N} \backslash \mathbb{N}_k)^{m-1} \\ \delta_{i i_2 \cdots i_m} = 0}} | a_{i i_2 \cdots i_m} | \text{ and } \sum_{i_2 \cdots i_m \in (\mathbb{N} \backslash \mathbb{N}_k)^{m-1}} | a_{i i_2 \cdots i_m} | .$$
Step 11. If Inequality (6) holds, then print "$\mathcal{A}$ is not a $\mathcal{H}$-tensor", and go to Step 5. Otherwise,
Step 12. Print "Whether A is a strong $\mathcal{H}$-tensor is not checkable", and replace $k_3$ by $k_3 + 1$. Go to Step 2.
```

4. Numerical Examples

Example 1. *Consider a tensor $\mathcal{A} = (a_{i_1 i_2 i_3})$, with order three and dimension six, defined as follows:*

$$\begin{aligned}
&a_{111} = 3.9,\ a_{112} = a_{121} = a_{122} = 1,\ a_{113} = a_{131} = a_{133} = a_{144} = a_{155} = 0.2,\\
&a_{222} = 5,\ a_{211} = a_{212} = a_{221} = a_{235} = a_{236} = 1,\ a_{214} = a_{215} = a_{232} = a_{256} = a_{266} = 0.2,\\
&a_{333} = 4,\ a_{334} = a_{343} = 0.5,\ a_{344} = 1,\ a_{311} = a_{322} = a_{331} = a_{326} = a_{352} = 0.2,\\
&a_{444} = 6.5,\ a_{433} = a_{434} = a_{443} = 1,\ a_{412} = a_{414} = a_{421} = a_{441} = 0.5,\\
&a_{555} = 6.3,\ a_{556} = 1,\ a_{512} = a_{515} = a_{521} = a_{523} = a_{551} = a_{565} = a_{566} = 0.5,\\
&a_{666} = 8,\ a_{655} = a_{656} = a_{665} = 1,\ a_{625} = a_{626} = a_{652} = a_{662} = 0.5
\end{aligned}$$

and other $a_{i_1 i_2 i_3} = 0$. By calculations, we have:

$$R_1(\mathcal{A}) = 4,\ R_2(\mathcal{A}) = 6,\ R_3(\mathcal{A}) = 3,\ R_4(\mathcal{A}) = 5,\ R_5(\mathcal{A}) = 4.5,\ R_6(\mathcal{A}) = 5,$$

$$\mathbb{N}_1 = \{1,2\},\ \mathbb{N}_2 = \{3,4\},\ \mathbb{N}_3 = \{5,6\},\ \mathbb{N} \backslash \mathbb{N}_1 = \{3,4,5,6\} = \mathbb{N}_2 \bigcup \mathbb{N}_3,$$

$$\begin{aligned}
&\mathbb{N}_1^{3-1} = \mathbb{N}_1^2 = \{11,12,21,22\},\ \mathbb{N}_2^{3-1} = \mathbb{N}_2^2 = \{33,34,43,44\},\ \mathbb{N}_3^{3-1} = \mathbb{N}_3^2 = \{55,56,65,66\},\\
&(\mathbb{N} \backslash \mathbb{N}_1)^{3-1} = (\mathbb{N} \backslash \mathbb{N}_1)^2 = \{33,34,35,36,43,44,45,46,53,54,55,56,63,64,65,66\},\\
&\mathbb{N}^{3-1} \backslash (\mathbb{N} \backslash \mathbb{N}_1)^{3-1} = \mathbb{N}^2 \backslash (\mathbb{N} \backslash \mathbb{N}_1)^2 = \{11,12,13,14,15,16,21,22,23,24,25,26,31,32,41,42,51,52,61,62\}.
\end{aligned}$$

$$\alpha_{\mathbb{N}_1}^{(1)} = \sum_{\substack{i_2 i_3 \in \mathbb{N}_1^2 \\ \delta_{1 i_2 i_3} = 0}} |a_{1 i_2 i_3}| = |a_{112}| + |a_{121}| + |a_{122}| = 1 + 1 + 1 = 3.$$

$$\alpha_{\mathbb{N}_1}^{(2)} = \sum_{\substack{i_2 i_3 \in \mathbb{N}_1^2 \\ \delta_{2 i_2 i_3} = 0}} |a_{2 i_2 i_3}| = |a_{211}| + |a_{212}| + |a_{221}| = 1 + 1 + 1 = 3.$$

$$\alpha_{\mathbb{N}_2}^{(3)} = \sum_{\substack{i_2 i_3 \in \mathbb{N}_2^2 \\ \delta_{3 i_2 i_3} = 0}} |a_{3 i_2 i_3}| = |a_{334}| + |a_{343}| + |a_{344}| = 0.5 + 0.5 + 1 = 2.$$

$$\alpha_{\mathbb{N}_2}^{(4)} = \sum_{\substack{i_2 i_3 \in \mathbb{N}_2^2 \\ \delta_{4 i_2 i_3} = 0}} |a_{4 i_2 i_3}| = |a_{433}| + |a_{434}| + |a_{443}| = 1 + 1 + 1 = 3.$$

$$\alpha_{\mathbb{N}_3}^{(5)} = \sum_{\substack{i_2 i_3 \in \mathbb{N}_3^2 \\ \delta_{5 i_2 i_3} = 0}} |a_{5 i_2 i_3}| = |a_{556}| + |a_{565}| + |a_{566}| = 1 + 0.5 + 0.5 = 2.$$

$$\alpha_{\mathbb{N}_3}^{(6)} = \sum_{\substack{i_2 i_3 \in \mathbb{N}_3^2 \\ \delta_{6 i_2 i_3} = 0}} |a_{6 i_2 i_3}| = |a_{655}| + |a_{656}| + |a_{665}| = 1 + 1 + 1 = 3.$$

$$\sum_{\substack{i_2 i_3 \in \mathbb{N}^2 \backslash (\mathbb{N} \backslash \mathbb{N}_1)^2 \\ \delta_{1 i_2 i_3} = 0}} |a_{1 i_2 i_3}| = |a_{112}| + |a_{121}| + |a_{122}| + |a_{113}| + |a_{131}| + 0 = 1 + 1 + 1 + 0.2 + 0.2 = 3.4.$$

$$\sum_{\substack{i_2 i_3 \in \mathbb{N}^2 \backslash (\mathbb{N} \backslash \mathbb{N}_1)^2 \\ \delta_{2 i_2 i_3} = 0}} |a_{2 i_2 i_3}| = |a_{211}| + |a_{212}| + |a_{221}| + |a_{214}| + |a_{215}| + |a_{232}| + 0 = 1 + 1 + 1 + 0.2 + 0.2 + 0.2 = 3.6.$$

$$\sum_{i_2 i_3 \in \backslash (\mathbb{N} \backslash \mathbb{N}_1)^2} |a_{1 i_2 i_3}| = |a_{133}| + |a_{144}| + |a_{155}| + 0 = 0.2 + 0.2 + 0.2 = 0.6.$$

$$\sum_{i_2 i_3 \in (\mathbb{N} \backslash \mathbb{N}_1)^2} |a_{2 i_2 i_3}| = |a_{235}| + |a_{236}| + |a_{256}| + |a_{266}| + 0 = 1 + 1 + 0.2 + 0.2 = 2.4.$$

$$\overline{\alpha}_{\mathbb{N}_1}^{(1)} = 1,\ \overline{\alpha}_{\mathbb{N}_1}^{(2)} = 3,\ \overline{\alpha}_{\mathbb{N}_2}^{(3)} = 1,\ \overline{\alpha}_{\mathbb{N}_2}^{(4)} = 2,\ \overline{\alpha}_{\mathbb{N}_3}^{(5)} = 2.5,\ \overline{\alpha}_{\mathbb{N}_3}^{(6)} = 2.$$

$$(|a_{111}| - \alpha_{\mathbb{N}_1}^{(1)}) \cdot (|a_{333}| - \alpha_{\mathbb{N}_2}^{(3)}) = (3.9 - 3) \times (4 - 2) = 1.8 > \overline{\alpha}_{\mathbb{N}_1}^{(1)} \cdot \overline{\alpha}_{\mathbb{N}_2}^{(3)} = 1 \times 1 = 1.$$

$$(|a_{111}| - \alpha_{\mathbb{N}_1}^{(1)}) \cdot (|a_{444}| - \alpha_{\mathbb{N}_2}^{(4)}) = (3.9 - 3) \times (6.5 - 3) = 3.15 > \overline{\alpha}_{\mathbb{N}_1}^{(1)} \cdot \overline{\alpha}_{\mathbb{N}_2}^{(4)} = 1 \times 2 = 2.$$

$$(|a_{111}| - \alpha_{\mathbb{N}_1}^{(1)}) \cdot (|a_{555}| - \alpha_{\mathbb{N}_3}^{(5)}) = (3.9 - 3) \times (6.3 - 2) = 3.87 > \overline{\alpha}_{\mathbb{N}_1}^{(1)} \cdot \overline{\alpha}_{\mathbb{N}_3}^{(5)} = 1 \times 2.5 = 2.5.$$

$$(|a_{111}| - \alpha_{\mathbb{N}_1}^{(1)}) \cdot (|a_{666}| - \alpha_{\mathbb{N}_3}^{(6)}) = (3.9 - 3) \times (8 - 3) = 4.5 > \overline{\alpha}_{\mathbb{N}_1}^{(1)} \cdot \overline{\alpha}_{\mathbb{N}_3}^{(6)} = 1 \times 2 = 2.$$

$$(|a_{222}| - \alpha_{\mathbb{N}_1}^{(2)}) \cdot (|a_{333}| - \alpha_{\mathbb{N}_2}^{(3)}) = (5 - 3) \times (4 - 2) = 4 > \overline{\alpha}_{\mathbb{N}_1}^{(2)} \cdot \overline{\alpha}_{\mathbb{N}_2}^{(3)} = 3 \times 1 = 3.$$

$$(|a_{222}| - \alpha_{\mathbb{N}_1}^{(2)}) \cdot (|a_{444}| - \alpha_{\mathbb{N}_2}^{(4)}) = (5 - 3) \times (6.5 - 3) = 7 > \overline{\alpha}_{\mathbb{N}_1}^{(2)} \cdot \overline{\alpha}_{\mathbb{N}_2}^{(4)} = 3 \times 2 = 6.$$

$$(|a_{222}| - \alpha_{\mathbb{N}_1}^{(2)}) \cdot (|a_{555}| - \alpha_{\mathbb{N}_3}^{(5)}) = (5 - 3) \times (6.3 - 2) = 8.6 > \overline{\alpha}_{\mathbb{N}_1}^{(2)} \cdot \overline{\alpha}_{\mathbb{N}_3}^{(5)} = 3 \times 2.5 = 7.5.$$

$$(|a_{222}|-\alpha_{\mathbb{N}_1}^{(2)})\cdot(|a_{666}|-\alpha_{\mathbb{N}_3}^{(6)})=(5-3)\times(8-3)=10>\overline{\alpha}_{\mathbb{N}_1}^{(2)}\cdot\overline{\alpha}_{\mathbb{N}_3}^{(6)}=3\times 2=6.$$

From Definition 4, we have $\mathcal{A}\in LDDT$. Furthermore,

$$(|a_{333}|-\alpha_{\mathbb{N}_2}^{(3)})\cdot(|a_{111}|-\sum_{\substack{i_2i_3\in\mathbb{N}^2\setminus(\mathbb{N}\setminus\mathbb{N}_1)^2\\ \delta_{1i_2i_3}=0}}|a_{1i_2i_3}|)=(4-2)\times(3.9-3.4)=1$$

$$>\overline{\alpha}_{\mathbb{N}_2}^{(3)}\cdot(\sum_{i_2i_3\in(\mathbb{N}\setminus\mathbb{N}_1)^2}|a_{1i_2i_3}|)=1\times 0.6=0.6,$$

$$(|a_{333}|-\alpha_{\mathbb{N}_2}^{(3)})\cdot(|a_{222}|-\sum_{\substack{i_2i_3\in\mathbb{N}^2\setminus(\mathbb{N}\setminus\mathbb{N}_1)^2\\ \delta_{2i_2i_3}=0}}|a_{2i_2i_3}|)=(4-2)\times(5-3.6)=2.8$$

$$>\overline{\alpha}_{\mathbb{N}_2}^{(3)}\cdot(\sum_{i_2i_3\in(\mathbb{N}\setminus\mathbb{N}_1)^2}|a_{2i_2i_3}|)=1\times 2.4=2.4,$$

$$(|a_{444}|-\alpha_{\mathbb{N}_2}^{(4)})\cdot(|a_{111}|-\sum_{\substack{i_2i_3\in\mathbb{N}^2\setminus(\mathbb{N}\setminus\mathbb{N}_1)^2\\ \delta_{1i_2i_3}=0}}|a_{1i_2i_3}|)=(6.5-3)\times(3.9-3.4)=1.75$$

$$>\overline{\alpha}_{\mathbb{N}_2}^{(4)}\cdot(\sum_{i_2i_3\in(\mathbb{N}\setminus\mathbb{N}_1)^2}|a_{1i_2i_3}|)=2\times 0.6=1.2,$$

$$(|a_{444}|-\alpha_{\mathbb{N}_2}^{(4)})\cdot(|a_{222}|-\sum_{\substack{i_2i_3\in\mathbb{N}^2\setminus(\mathbb{N}\setminus\mathbb{N}_1)^2\\ \delta_{2i_2i_3}=0}}|a_{2i_2i_3}|)=(6.5-3)\times(5-3.6)=4.9$$

$$>\overline{\alpha}_{\mathbb{N}_2}^{(4)}\cdot(\sum_{i_2i_3\in(\mathbb{N}\setminus\mathbb{N}_1)^2}|a_{2i_2i_3}|)=2\times 2.4=4.8,$$

$$(|a_{555}|-\alpha_{\mathbb{N}_3}^{(5)})\cdot(|a_{111}|-\sum_{\substack{i_2i_3\in\mathbb{N}^2\setminus(\mathbb{N}\setminus\mathbb{N}_1)^2\\ \delta_{1i_2i_3}=0}}|a_{1i_2i_3}|)=(6.3-2)\times(3.9-3.4)=2.15$$

$$>\overline{\alpha}_{\mathbb{N}_3}^{(5)}\cdot(\sum_{i_2i_3\in(\mathbb{N}\setminus\mathbb{N}_1)^2}|a_{1i_2i_3}|)=2.5\times 0.6=1.5,$$

$$(|a_{555}|-\alpha_{\mathbb{N}_3}^{(5)})\cdot(|a_{222}|-\sum_{\substack{i_2i_3\in\mathbb{N}^2\setminus(\mathbb{N}\setminus\mathbb{N}_1)^2\\ \delta_{2i_2i_3}=0}}|a_{2i_2i_3}|)=(6.3-2)\times(5-3.6)=6.02$$

$$>\overline{\alpha}_{\mathbb{N}_3}^{(5)}\cdot(\sum_{i_2i_3\in(\mathbb{N}\setminus\mathbb{N}_1)^2}|a_{2i_2i_3}|)=2.5\times 2.4=6,$$

$$(|a_{666}|-\alpha_{\mathbb{N}_3}^{(6)})\cdot(|a_{111}|-\sum_{\substack{i_2i_3\in\mathbb{N}^2\setminus(\mathbb{N}\setminus\mathbb{N}_1)^2\\ \delta_{1i_2i_3}=0}}|a_{1i_2i_3}|)=(8-3)\times(3.9-3.4)=2.5$$

$$>\overline{\alpha}_{\mathbb{N}_3}^{(6)}\cdot(\sum_{i_2i_3\in(\mathbb{N}\setminus\mathbb{N}_1)^2}|a_{1i_2i_3}|)=2\times 0.6=1.2,$$

$$(|a_{666}|-\alpha_{\mathbb{N}_3}^{(6)})\cdot(|a_{222}|-\sum_{\substack{i_2i_3\in\mathbb{N}^2\setminus(\mathbb{N}\setminus\mathbb{N}_1)^2\\ \delta_{2i_2i_3}=0}}|a_{2i_2i_3}|)=(8-3)\times(5-3.6)=7$$

$$>\overline{\alpha}_{\mathbb{N}_3}^{(6)}\cdot(\sum_{i_2i_3\in(\mathbb{N}\setminus\mathbb{N}_1)^2}|a_{2i_2i_3}|)=2\times 2.4=4.8.$$

These imply that $\mathcal{A}$ satisfies all the conditions of Theorem 1. Therefore, $\mathcal{A}$ is an $\mathcal{H}$-tensor.

5. Conclusions

In this paper, several practical criteria for identifying $\mathcal{H}$-tensors were given. These criteria only depend on the self-constituent elements of the tensors. Therefore, they are easy to implement. Based on the above criteria, we also gave the corresponding algorithm to determine the $\mathcal{H}$-tensor.

Author Contributions: M.L. and H.S. presented the idea and algorithms as supervisors. P.L. contributed to the preparation of the paper. G.H. contributed to the implementation of the algorithm. All authors have read and agreed to the submitted version of the manuscript.

Funding: This research was funded by the Jilin Province Department of Education, Science and Technology research project under Grant JJKH20220041KJ, the 13th Five-Year Plan of Educational Science in Jilin Province under Grant GH19057, and the National Natural Science Foundations of China (11171133).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their gratitude to the Editor for his assistance, as well as the anonymous respected reviewers for providing insightful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, G.; Qi, L.; Yu, G. The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory. *Numer. Linear Algebra Appl.* **2013**, *20*, 1001–1029. [CrossRef]
2. Liu, J.; Musialski, P.; Wonka, P.; Ye, J. Tensor completion for estimating missing valuesin visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *35*, 208–220. [CrossRef] [PubMed]
3. Moakher, M. On the averaging of symmetric positive-definite tensors. *J. Elast.* **2006**, *82*, 273–296. [CrossRef]
4. Nikias, C.L.; Mendel, J.M. Signal processing with higher-order spectra. *IEEE Signal Process. Mag.* **1993**, *10*, 10–37. [CrossRef]
5. Ni, Q.; Qi, L.; Wang, F. An eigenvalue method for the positive definiteness identification problem. *IEEE Trans. Automat. Control* **2008**, *53*, 1096–1107. [CrossRef]
6. Michael, N.; Qi, L.; Zhou, G. Finding the largest eigenvalue of a non-negative tensor. *SIAM J. Matrix Anal. Appl.* **2010**, *31*, 1090–1099.
7. Oeding, L.; Ottaviani, G. Eigenvectors of tensors and algorithms for Waring decomposition. *J. Symb. Comput.* **2013**, *54*, 9–35. [CrossRef]
8. Yang, Y.; Yang, Q. Singular values of nonnegative rectangular tensors. *Front. Math. China* **2011**, *6*, 363–378. [CrossRef]
9. Brachat, J.; Comon, P.; Mourrain, B.; Tsigaridas, E. Symmetric tensor decomposition. *Linear Algebra Appl.* **2010**, *433*, 1851–1872. [CrossRef]
10. Cichocki, A.; Zdunek, R.; Phan, A.H.; Amari, S.I. *Nonnegative Matrix and Tensor Factorizations*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 25, pp. 1–3.
11. Tartar, L. Mathematical Tools for Studying Oscillations and Concentrations: From Y oung Measures to H-Measures and Their Variants. In *Multiscale Problems in Science and Technology*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 1–84.
12. Zhang, L.; Qi, L.; Zhou, G. $\mathcal{M}$-tensors and some applications. *SIAM J. Matrix Anal. Appl.* **2014**, *35*, 437–452. [CrossRef]
13. De Lathauwer, L.; De Moor, B.; Vandewalle, J. A multilinear singular value decomposition. *SIAM J. Matrix Anal.* **2000**, *21*, 1253–1278. [CrossRef]
14. Qi, L. Eigenvalues of a real supersymetric tensor. *J. Symb. Comput*. **2005**, *40*, 1302–1324. [CrossRef]
15. Yang, Y.; Yang, Q. Fruther results for Perron-Frobenius theorem for nonnegative tensors. *SIAM J. Mayrix Anal. Appl*. **2010**, *31*, 2517–2530. [CrossRef]
16. Li, C.Q.; Li, Y.T.; Xu, K. New eigenvalue inclusion sets for tensor. *Numer. Linear Algebra Appl.* **2014**, *21*, 39–50. [CrossRef]
17. Kolda, T.G.; Mayo, J.R. Shifted power method for computing tensor eigenpairs. *SIAM J. Matrix Anal. Appl.* **2011**, *32*, 1095–1124. [CrossRef]
18. Lim, L.H. Singular values and eigenvalues of tensors: A variational approach. In Proceedings of the IEEE International Workshop on Computational Advances in Multisensor Adaptive Processing, Puerto Vallarta, Mexico, 13–15 December 2005; pp. 129–132.
19. Qi, L. Eigenvalues and invariants of tensors. *J. Math. Anal. Appl.* **2007**, *325*, 1363–1377. [CrossRef]
20. Qi, L.; Wang, F.; Wang, Y. Z-eigenvalue methods for a global polynomial optimization problem. *Math. Program.* **2009**, *118*, 301–316. [CrossRef]
21. Ni, G.; Qi, L.; Wang, F.; Wang, Y. The degree of the *E*-characteristic polynomial of an even order tensor. *J. Math. Anal. Appl.* **2007**, *329*, 1218–1229. [CrossRef]

22. Ding, W.; Qi, L.; Wei, Y. $\mathcal{M}$-tensors and nonsingular $\mathcal{M}$-tensors. *Linear Algebra Appl.* **2013**, *439*, 3264–3278. [CrossRef]
23. Kannana, M.R.; Mondererb, N.S.; Bermana, A. Some properties of strong $\mathcal{H}$-tensors and general $\mathcal{H}$-tensors. *Linear Algebra Appl.* **2015**, *476*, 42–55. [CrossRef]
24. Hu, S.; Li, G.; Qi, L.; Song, Y. Finding the maximum eigenvalue of essentially nonnegative symmetric tensors via sum of squares programming. *J. Optim. Theory Appl.* **2013**, *158*, 713–738. [CrossRef]
25. Hu, S.; Li, G.; Qi, L. A tensor analogy of Yuans theorem of the alternative and polynomial optimization with sign structure. *J. Optim. Theory Appl.* **2016**, *168*, 446–474. [CrossRef]
26. Cui, C.F.; Dai, Y.H.; Nie, J.W. All real eigenvalues of symmetric tensors. *SIAM J. Matrix Anal. Appl.* **2014**, *35*, 1582–1601. [CrossRef]
27. Li, C.Q.; Wang, F.; Zhao, J.X.; Li, C.; Zhu, Y.; Li, Y. Criterions for the positive definiteness of real supersymmetric tensors. *J. Comput. Appl. Math.* **2014**, *255*, 1–14. [CrossRef]

symmetry

MDPI

Article

A New Meta-Heuristics Data Clustering Algorithm Based on Tabu Search and Adaptive Search Memory

Youseef Alotaibi

Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia; yaotaibi@uqu.edu.sa

Abstract: Clustering is a popular data analysis and data mining problem. Symmetry can be considered as a pre-attentive feature, which can improve shapes and objects, as well as reconstruction and recognition. The symmetry-based clustering methods search for clusters that are symmetric with respect to their centers. Furthermore, the K-means (K-M) algorithm can be considered as one of the most common clustering methods. It can be operated more quickly in most conditions, as it is easily implemented. However, it is sensitively initialized and it can be easily trapped in local targets. The Tabu Search (TS) algorithm is a stochastic global optimization technique, while Adaptive Search Memory (ASM) is an important component of TS. ASM is a combination of different memory structures that save statistics about search space and gives TS needed heuristic data to explore search space economically. Thus, a new meta-heuristics algorithm called (MHTSASM) is proposed in this paper for data clustering, which is based on TS and K-M. It uses TS to make economic exploration for data with the help of ASM. It starts with a random initial solution. It obtains neighbors of the current solution called trial solutions and updates memory elements for each iteration. The intensification and diversification strategies are used to enhance the search process. The proposed MHTSASM algorithm performance is compared with multiple clustering techniques based on both optimization and meta-heuristics. The experimental results indicate the superiority of the MHTSASM algorithm compared with other multiple clustering algorithms.

Keywords: data clustering; Tabu Search; K-means; adaptive memory; meta-heuristics

Citation: Alotaibi, Y. A New Meta-Heuristics Data Clustering Algorithm Based on Tabu Search and Adaptive Search Memory. *Symmetry* **2022**, *14*, 623. https://doi.org/10.3390/sym14030623

Academic Editor: Kuo-Hui Yeh

Received: 3 March 2022
Accepted: 18 March 2022
Published: 20 March 2022

1. Introduction

Clustering can be considered as a key problem in data analysis and data mining [1]. It tries to group similar data objects into the sets of disjoint classes or clusters [2,3]. Symmetry can be considered as a pre-attentive feature that can improve the shapes and objects as well as reconstruction and recognition. Furthermore, the symmetry-based clustering methods search for clusters that are symmetric with respect to their centers. As each cluster has the point symmetry property, an efficient point symmetry distance (PSD) was proposed by Su and Chou [4] to support partitioning the dataset into the clusters. They proposed a symmetry-based K-means (SBKM) algorithm that can give symmetry with minimum movement [5].

The literature shows that there are many techniques to solve clustering problems. They are basically categorized as the hierarchical clustering algorithm and the partitional clustering algorithm [6,7].

The hierarchical clustering algorithms mainly generate clusters hierarchies signified in a tree structure called a dendrogram. It can be successively proceeded throughout either splitting larger clusters or merging smaller clusters into larger ones. The clusters identify as the connected components in a tree. Hence, the output of hierarchical clustering algorithms is a tree of clusters [8–10].

However, partitional clustering algorithms attempt to treat a dataset as one cluster and try to decompose it into disjoint clusters. As partitional clustering does not need to make a tree structure, it can be considered to be easier than hierarchy clustering. The partitional algorithms are restricted with a criterion function while decomposing the dataset, and thus this function leads to minimizing dissimilarity in each cluster and maximizing similarity in data objects of each cluster [11].

The K-means (K-M) algorithm is considered to be one of the most common algorithms of partitional clustering [12–14]. However, it has some disadvantages, such as specifying the number of k clusters, as it is based on practical experience and knowledge. The main K-M problems are its sensitivity to initialization and becoming trapped in local optima [15,16]. Therefore, meta-heuristics algorithms are used to escape from these problems.

Meta-heuristics is an iterative generation processes that can guide a heuristic algorithm by different responsive concepts for exploring and exploiting search space [17]. It is a class of approximate methods that can attack hard combinatorial problems when the classical optimization methods fail. It allows for the creation of new hybrid methods by combining different concepts [18,19].

Tabu Search (TS) is considered to be one of the meta-heuristics algorithms based on memory objects to make economic exploitation and exploration for search space. Tabu list (TL) is one of the main memory elements of TS. It is a list that contains recently visited solutions by TS. It is used by TS in order to avoid re-entering pre-visited regions [20,21].

This paper aims to propose a new meta-heuristics Tabu Search with an Adaptive Search Memory algorithm (MHTSASM) for data clustering, which is based on TS and K-M. MHTSASM is an advanced TS that uses responsive and intelligent memory. The proposed MHTSASM algorithm uses multiple memory elements. For example, it uses the elite list (EL) memory element of TS to store the best solutions and Adaptive Search Memory (ASM) to store information about features of each solution. The ASM partitions each feature range into p partitions and stores information about these partitions in each center. There are two types of ASM used in the proposed algorithm: (1) ASMv stores number of visits; and (2) ASMu stores the number of improvement updates. In addition, there are two TS strategies used in the proposed algorithm: (1) the intensification strategy, which allows MHTSASM to focus on the best pre-visited regions; and (2) the diversification strategy, which allows MHTSASM to explore un-visited regions to ensure good exploration for search space.

The results of MHTSASM are specifically compared with other optimization and metaheuristics algorithms. The performance of MHTSASM is compared with several variable neighborhood searches, such as Variable Neighborhood Search (VNS), VNS+, and VNS-1 algorithms [22], Simulated Annealing (SA) [23], and global K-means (G1) [24]. Additionally, it is compared with hybrid algorithms, such as a new hybrid algorithm based on particle swarm optimization with K-harmonic means (KHM) [25], and an ant colony optimization clustering algorithm (ACO) with KHM [26,27].

The remainder of this paper is organized as follows: Section 2 presents the background of the proposed MHTSASM clustering algorithm. Section 3 presents the proposed MHTSASM clustering algorithm. Section 4 illustrates experimental results. Finally, the conclusion is presented in Section 5.

2. Background

2.1. K-Means (K-M) Clustering Algorithm

The K-M is a common clustering algorithm for data mining used in many real life applications, such as healthcare, environment and air pollution, and industry data. It outputs k centers that partition input points into k clusters [12–14]. However, it has some disadvantages, such as specifying the number of k clusters, as it is based on the practical experience and knowledge. An example of how to select the k number is shown in Figure 1. It has two clusterings, where the dark crosses specify the k centers. There are five k values used on the left. In addition, there are many centers that could be used on the right, even though one center can sufficiently represent the data [15]. The main K-M problems are its sensitivity to initialization and getting trapped in local optima [16]. Therefore, meta-heuristics algorithms are used to escape from these problems.

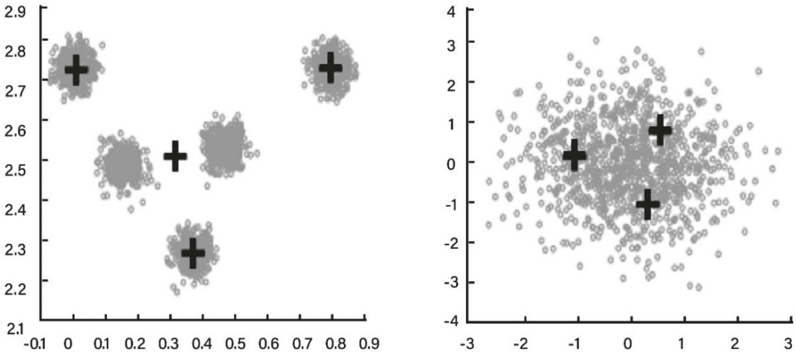


Figure 1. An example of the K-means clustering algorithm.

The K-M calculates the center of each cluster as the mean value of points that belong to the cluster. It works as follows: let D denotes the entire dataset, n the number of data objects, and d the number of features Rd, i.e., $D = \{x_1, x_2, \ldots, x_n\}$, and we want to partition set X into K partitions. K-M tries to obtain the best set of centers $C = \left\{C^1, \ldots, C^k\right\}$. Let C^j denote class j with cardinality $|S_j|$; then we can calculate C^j, j = 1, ... , K according to the following equation:

$$C^j = \frac{1}{|S_j|} \sum_{x \in S_j} x \tag{1}$$

In order to measure the goodness of different sets of centers, objective function f(C) can be calculated for each set of centers using the minimum sum of squares [10,11], defined by the following:

$$f(C) = \frac{1}{n} \sum_{i=1}^{n} \min_{j} d\,(C_j,\ x_i) \tag{2}$$

where $d(C_j,\ x_i)$ corresponds to the squared Euclidean distance between the center of class C and the point x_i.

K-M is known for its low computational requirements and ease of implementation. The main K-M problems are its sensitivity to initialization and becoming trapped in local optima [16,28]. Therefore, the meta-heuristics algorithms are used to escape from these problems.

2.2. Meta-Heuristics Clustering Algorithm

There are many meta-heuristics methods that can be used with K-M in order to increase its efficiency and treat its problems. The literature shows that there are several methods proposed in order to compete local optima problem of K-M, such as SA, TS, and the genetic algorithm (GA) [23].

A hybrid algorithm that uses GA and K-M (GKA) was proposed in [29,30]. GKA uses GA as a search operator, as it finds a globally optimal partition [31,32].

In addition, a hybrid algorithm was proposed in [33] to solve the problems of K-M based on GA and the weighted K-M algorithm. Moreover, another hybrid algorithm using K-M and SA was proposed in [34].

A comparison of implementation complexity and computational cost between the most common clustering algorithms was summarized, as shown Table 1 [16,23,35–37].

Table 1. Comparison between different optimization and metaheuristics algorithms [2,22,38–40].

Algorithm Name	Implementation Complexity Level			Computational Cost Level		
	High	Medium	Low	High	Medium	Low
K-M			✓			✓
TS			✓			✓
GA		✓		✓		
SA		✓			✓	
Algorithm 1		✓			✓	
J-M+			✓			✓
VNS		✓			✓	
VNS-1		✓			✓	
G1		✓				✓
G2		✓				✓
KHM		✓			✓	
PSO		✓			✓	
PSOKHM	✓				✓	
ACO	✓				✓	
ACOKHM	✓				✓	
Proposed MHTSASM Approach			✓			✓

2.2.1. Tabu Search (TS) Clustering Algorithm

TS is considered to be one of the meta-heuristics algorithms that is based on memory objects to make economic exploitation and exploration for search space. Tabu list (TL) is one of the main memory elements of TS. It is a list that contains recently visited solutions by TS. It is used by TS in order to avoid re-entering pre-visited regions.

Memory Elements

TS is considered to be one of the meta-heuristic algorithms. It is used to solve hard optimization problems of which classical methods often encounter great difficulty [41]. It depends on memory structures rather than memoryless methods based on semi random processes implementing a form of sampling. The MHTSASM is an advanced TS that uses responsive and intelligent memory. It uses multiple memory elements such as EL, TL, and ASM in order to minimize search time and focus on promising regions.

EL: Elite list stores best visited solutions.

TL: Tabu list stores recently visited solutions.

ASMv: Feature partitions visitation matrix. It is a matrix that partitions each feature into p partitions, and it is stored as a matrix of size K × p × d. It stores the number of visits of each partition for all generated centers.

ASMu: Feature improvement update matrix. It has the same size of ASMv and stores the number of obtained improvements of objective functions while updating the centers.

MHTSASM uses both explicit and attributive memories. Explicit memory records complete solutions such as EL and TL. Attributive memory stores information about features of each solution such as ASMv and ASMu. MHTSASM uses memory elements to guide intensification and diversification strategies.

Center Adjustment

The center adjustment method as shown in Algorithm 1 is used to re-compute the centers according to the data objects related to each center. It re-assigns each data object x to center C^j, j = 1, . . . , K with minimum Euclidean distance between center and data object. Applying K-M, a new set of centers according to mean values of related objects to each center is computed. Equations (3) and (4) are used in Algorithm 1.

$$j^* = \arg \min_{j = 1,\ldots,k} \|x^i - \widetilde{C}^{(j)}\| \tag{3}$$

$$\widetilde{C}^{(j)} = \frac{1}{|S_j|} \sum\nolimits_{x \in S_j} x \tag{4}$$

Algorithm 1: Center Adjust(K, $\widetilde{C}$, d, D)

```
1.  Set $S_j$, j = 1, . . . , K to be empty.
2.  Repeat (2.1–2.2) steps for i = 1, . . . , n.
    2.1. Compute the Euclidean distance between data object xi and each center $\widetilde{C}^{(j)}$, j = 1, . . . , K.
    2.2. Add $x^i$ to $S_j*$ where
Compute Equation (3)
    Hence, $S_{j*} = S_{j*} \cup x^i$
3.  Compute a new set of centers $\widetilde{C} = \left\{\widetilde{C}^{(1)}, \ldots, \widetilde{C}^{(k)}\right\}$; each center $\widetilde{C}^{(j)} \in \widetilde{C}$ is computed
    according to the following equation:
Compute Equation (4)
4.  Return $\widetilde{C}^{(j)}$, j = 1, . . . , K.
```

3. The Proposed MHTSASM Algorithm

In this paper, the MHTSASM algorithm is proposed for data clustering, which is based on TS and K-M. Figure 2 shows the proposed MHTSASM algorithm framework. The MHTSASM starts with a random initial solution. It obtains neighbors of the current solution called trial solutions and updates memory elements for each iteration. Furthermore, it uses intensification and diversification strategies guided by memory elements to enhance the search process. Thus, it is a high level TS algorithm with long term memory elements. In the following subsections, the main components of MHTSASM are explained.

3.1. Intensification and Diversification Strategies

The TS has two highly important components called (1) intensification strategy; and (2) diversification strategy. The MHTSASM uses both intensification and diversification strategies guided by ASM. Intensification tries to immigrate to best solutions. It can recruit a return into striking regions in order to explore them more systematically. Furthermore, intensification techniques require saving elite solutions to be able to examine their neighbors, and thus the MHTSASM records elite solutions in EL. The MHTSASM may use the responsive memory ASMu in order to keep track of best partitions of each feature in the search space.

The intensification method as shown in Algorithm 2 is guided by ASMu. It selects a random number of features from d features. The MHTSASM selects a partition with the highest updating improvement value for the current center for each selected feature. Each selected feature value is updated with a random value in range between the current feature value and the selected partition. The memory elements are updated, and the updated set of centers is returned. Algorithm 2 states formally the steps of the intensification process.

On the other hand, the diversification strategy can develop the exploration process in order to inspect the non-visited regions and create a new solution. It can improve the exploration for search space. The MHTSASM applies diversification by choosing partitions from ASMv with a minimum number of visits.

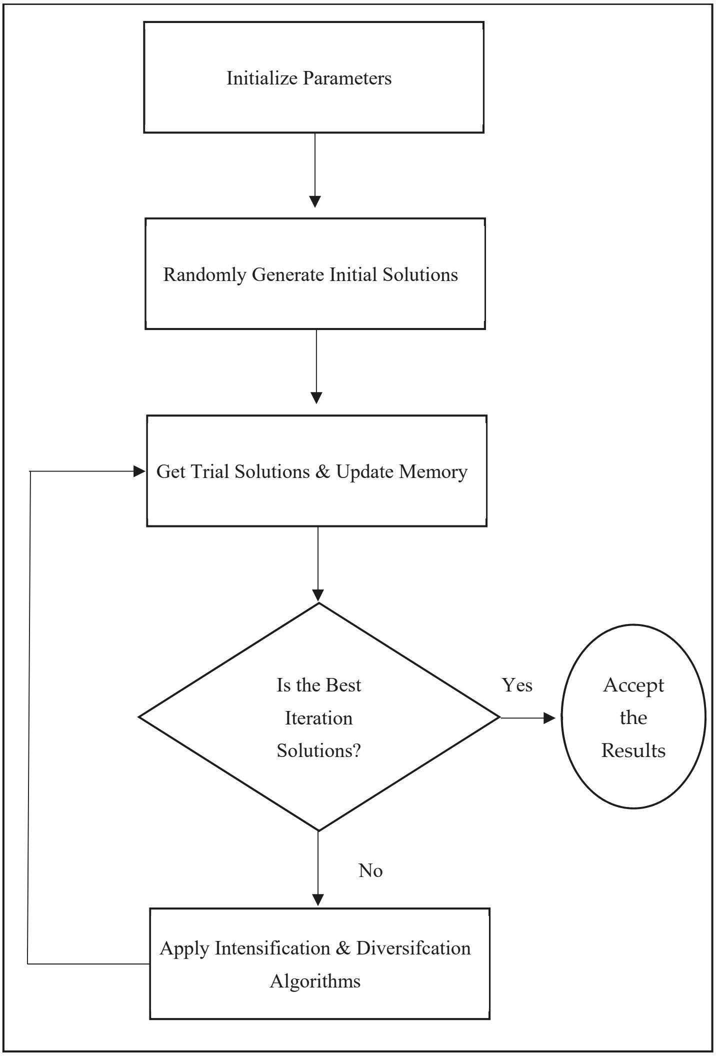

Initialize Parameters
Randomly Generate Initial Solutions
Get Trial Solutions & Update Memory
Is the Best Iteration Solutions?
Yes
Accept the Results
No
Apply Intensification & Diversifcation Algorithms

Figure 2. The proposed MHTSASM algorithm framework.

Algorithm 2: Intensification Algorithm $\left(\delta,\ \widetilde{C},\ ASM_u,\ ASM_v,\ d\right)$

```
1.  For each center in $\widetilde{C}$, repeat the following steps (1.1–1.4).
    1.1.  Select $\delta$ features with highest updating improvement value in $ASM_u$ from d
          data features.
    1.2.  Select a partition from p partitions for each selected feature with the highest
          updating improvement value for current center in $ASM_u$.
    1.3.  Update the value of the selected feature by computing a random value in the range
          between the current value and the selected partition value.
    1.4.  Update memory elements $ASM_v$.
2.  Return the updated centers $\widetilde{C}$, $ASM_v$.
```

The diversification method as shown in Algorithm 3 tends to explore unvisited regions guided by ASMv. It selects a random number of features from d features. Next, the MHTSASM selects a partition with a minimum number of visits for the current center for the selected features. Each feature value is updated with a random value in the range of the selected partition depending on the ASMv information. The memory elements are updated, and the updated set of centers is returned. Algorithm 3 states formally the steps of the diversification process.

Algorithm 3: Diversification Method $\left(\delta, \widetilde{C}, ASM_v, d\right)$

```
1.    For each center in $\widetilde{C}$, repeat the following steps (1.1–1.4).
      1.1.    Select $\delta$ features from features with the least number of visits in $ASM_v$ from d
              data features.
      1.2.    Select a partition from p partitions for each selected feature with the least number of
              visits for the current center in $ASM_v$.
      1.3.    Update the value of the selected feature by computing a random value in the range
              between the current value and the selected partition value.
      1.4.    Update memory elements $ASM_v$.
2.    Return the updated centers $\widetilde{C}$, $ASM_v$.
```

3.2. Trial Solutions Generation Algorithm

The trial solutions generation method as shown in Algorithm 4 generates μ trial solutions near the current set of centers C. It may select the random features for each center and then generate a new random value that belongs to the range of the current value partition. It makes a thorough search for neighbors of the current set of centers C. Algorithm 4 states formally the steps of the trial solutions generation process.

Algorithm 4: Trial Solutions Method (C, μ, ASM_v, d)

```
1.    Repeat the following steps (1.1–1.3) for j = 1, . . . , $\mu$.
      1.1.    Set $\widetilde{C}^{(j)}$ equal to the input centers C.
      1.2.    Select $\eta$ features randomly from d data features.
      1.3.    Repeat steps (1.3.1–1.3.2) for m = 1, . . . , $\eta$.
              1.3.1.  Update the values of the selected features in $\widetilde{C}^{(j)}$ by generating new values
                      randomly.
              1.3.2.  Update the memory elements $ASM_v$.
2.    Return the updated set of centers $\widetilde{C}^{(j)}$, j = 1, . . . , $\mu$, $ASM_v$.
```

3.3. Refinement Method

The refinement method as shown in Algorithm 5 is used to examine the neighbors of the best solutions that have been found so far and return with the better solution if it exists in their neighborhood. It may call the trial solution (Algorithm 4) in order to find the neighbors of the current solution. If a better solution is found, it will call the trial solution (Algorithm 4) again for a new solution; otherwise, it will break. Algorithm 5 states formally the steps of the refinement method.

Algorithm 5: Refinement Method (C, μ, ASM_v, ASM_u, EL, d)

```
1.    Apply Algorithm 4 to generate $\mu$ centers $\widetilde{C}$ near the set of centers C.
2.    Update the memory elements.
3.    If an improvement achieved, then go to step 1; otherwise, go to step 4.
4.    Return C, $ASM_v$, $ASM_u$, EL.
```

3.4. The Proposed MHTSASM Algorithm

The MHTSASM cluster algorithm is proposed in this paper. It is an advanced TS that uses responsive and intelligent memory, and it uses multiple memory elements, such as the EL memory element of TS, in order to store the best solutions, and the ASM in order to store information about features of each solution. It may use two important TS strategies: intensification and diversification. It starts with initial random centers and then applies the trail solutions (Algorithm 4) in order to generate trial solutions near initial centers. The parameter variables are used to determine if intensification or diversification is needed.

If the intensification strategy is needed, the intensification (Algorithm 2) is applied to enhance the set of centers using ASM and EL. If the centers are stuck and gain no improvement, diversification (Algorithm 3) is used to make exploration for regions that have not been visited yet. However, if termination conditions are satisfied, then refinement is carried out (Algorithm 5), where the refinement method examines the neighbors of the best solution that has been found so far. If there is no improvement, then the process is terminated. Algorithm 6 states formally the steps of the advanced MHTSASM algorithm.

Algorithm 6: MHTSASM Algorithm

```
1.  Initialization. Set values for the number of clusters K, the number of neighborhood trials $\mu$,
    and set the memory elements TL, $ASM_v$, $ASM_u$, and EL to be empty.
2.  Generate K random initial centers C.
3.  Evaluate the initial centers C and update the memory elements TL and EL.
4.  Main Loop.
    4.1.  Generate $\mu$ trail solutions $\widetilde{C}^{(j)}$, j = 1, . . . , $\mu$ applying Algorithm 4.
    4.2.  Repeat the following steps (4.2.1–4.2.2) for j = 1, . . . , $\mu$.
          4.2.1.  Adjust set of centers $\widetilde{C}^{(j)}$ applying Algorithm 1; compute the objective
                  function $f(\widetilde{C}^{(j)})$ applying Equation (2).
          4.2.2.  If objective function of $f(\widetilde{C}^{(j)})$ is not better than f(C), then go to Step 4.2.1;
                  otherwise set C equal to $\widetilde{C}^{(j)}$ and update $ASM_u$.
    4.3.  Update the memory elements TL and EL.
5.  Termination. If the termination conditions are satisfied then go to Step 8. Otherwise, go to
    Step 6.
6.  Intensification. If an intensive solution is needed, then apply Algorithm 3 to generate new
    centers C. Update the memory elements and go to Step 4.
7.  Diversification. If a diverse solution is needed, then apply Algorithm 2 to generate new
    centers C. Update the memory elements and go to Step 4.
8.  Final Refinement. Apply Algorithm 5 to refine the best solution found so far.
```

4. Numerical Experiments

4.1. Dataset

There are several numerical experiments for small and medium dataset sizes that were done in order to validate the clustering algorithm effectiveness. In other words, there were three clustering test problems used in order to evaluate the proposed MHTSASM algorithm with others clustering techniques, such as K-M, TS, GA, and SA.

Firstly, there are three different standard test problems that were considered to compare the proposed algorithm with the other heuristics and meta-heuristics, such as the K-M algorithm [42], the TS method, GA, the SA method, and the non-smooth optimization technique for the minimum sum for the squares clustering problems (Algorithm 1) [43]. The German towns clustering test, which uses Cartesian coordinates, which has 59 towns and 59 records with two attributes, was used. Secondly, another set including 89 postal zones in Bavaria, which includes 89 records and 3 attributes, was used. Thirdly, another set including 89 postal zones in Bavaria, which includes 89 records and 4 attributes, was used. The results for these datasets are presented in Section 4.2.1, First Clustering Test Problems Situation.

Secondly, another two test datasets were used to compare the proposed MHTSASM algorithm with K-M algorithm, j-means (J-M+) algorithms [25], and the variable neighborhood search, such as VNS, VNS+, and VNS-1 algorithms [22]. In addition, we compared MHTSASM with the Global K-means (G1) [24], Fast global K-means (G2) algorithm [38,39], and Algorithm 1 [40,44]. These datasets are called (1) the first Germany postal zones dataset; and (2) the Fisher's iris dataset that includes 150 instances and 4 attributes [45–47] used in order to evaluate the proposed MHTSASM algorithm with others clustering techniques, such as K-M, J-M+, VNS, VNS+, VNS-1, G1, and G2. The results for these dataset is presented in Section 4.2.2, Second Clustering Test Problems Situation.

Thirdly, another five datasets called Iris, Glass, Cancer, Contraceptive Method Choice (CMC) and Wine as the examples of low, medium, and high dimensions data, respectively, were used to compare the proposed MHTSASM algorithm with K-harmonic means (KHM), particle swarm optimization (PSO), hybrid clustering method based on KHM, and PSO, called the PSOKHM algorithm [25] ACO and the hybrid clustering method based on ACO and KHM called the ACOKHM algorithm [26,27]. The results for these dataset is presented in Section 4.2.3, Third Clustering Test Problems Situation.

Table 2 summarizes the characteristics of these datasets [47]. Furthermore, there are several methods called (1) KHM; (2) PSO; and (3) PSOKHM algorithms [25], which were selected in order to compare their performances. In addition, other method called (1) KHM; (2) ACO; and (3) ACOKHM algorithms [27,28] were selected for the same purpose.

Table 2. Datasets characteristics.

Data Set Name	No. of Classes	No. of Features	Data Set Size	Classes Size in Parentheses
Iris	3	4	150	(50, 50, 50)
Glass	6	9	214	(70, 17, 76, 13, 9, 29)
Cancer	2	9	683	(444, 239)
CMC	3	9	1473	(629, 334, 510)
Wine	3	13	178	(59, 71, 48)

Moreover, there are two characteristics for each dataset that can positively affect the clustering algorithms' performances: (1) the instances which are the data records numbers; and (2) the features which are the data attributes numbers. The list of parameter values for the proposed MHTSASM algorithm is presented in Table 3 [28].

Table 3. Parameters setting.

Parameters	Definition	Value
El	Largest number of solution stored in elite list	5
$\mathrm{ASM_v}$	Number of visits of each partition	9
$\mathrm{ASM_u}$	Number of visit of each partition with improvement	300

4.2. Clustering Test Problems Situation Results

Tables 4 and 5 report the best value for the global minimum. The values are given as $\mathrm{nf(x^*)}$, where $\mathrm{x^*}$ is the local minimizer, and n is the instances number [28]. Moreover, the error E percentage of every algorithm is shown and calculated using Equation (5).

$$\mathrm{E} = \frac{\bar{\mathrm{f}} - \mathrm{f_{opt}}}{\mathrm{f_{opt}}} * 100 \tag{5}$$

where $\bar{\mathrm{f}}$ represents the algorithm finding solution, and fopt represents the best known solution.

Table 4. Results of first problems situation.

K	Dataset Name	f_{opt}	K-M	TS	GA	SA	Algorithm 1	MHTSASM
2	German towns	0.121426×10^6	0.00	0.00	0.00	0.00	0.00	0.00
	first Bavarian postal zones	0.60255×10^{12}	7.75	0.00	0.00	0.00	0.00	0.00
	second Bavarian postal zones	0.199080×10^{11}	144.25	0.00	144.25	144.25	144.25	144.25
3	German towns	0.77009×10^5	1.45	0.00	0.00	0.29	0.29	0.00
	first Bavarian postal zones	0.29451×10^{12}	23.48	23.48	23.48	23.48	0.00	0.00
	second Bavarian postal zones	0.173987×10^{11}	106.79	0.00	0.00	77.77	0.00	0.00
4	German towns	0.49601×10^5	0.55	0.00	0.00	0.00	0.00	0.00
	first Bavarian postal zones	0.10447×10^{12}	166.88	18.14	0.00	0.39	0.00	0.00
	second Bavarian postal zones	0.755908×10^{10}	303.67	0.00	0.00	9.13	0.00	0.00
5	German towns	0.39453×10^5	2.75	0.00	0.15	0.15	0.15	0.00
	first Bavarian postal zones	0.59762×10^{11}	335.32	33.35	0.00	40.32	0.00	0.00
	second Bavarian postal zones	0.540379×10^{10}	446.13	15.76	15.76	18.72	0.00	0.00

Table 5. Results of second problems situation.

K	Dataset Name	f_{opt}	K-M	J-M+	VNS	VNS-1	G1	G2	Algorithm 1	MHTSASM
2	first Germany postal zones	0.60255×10^{12}	7.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	152.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
3	first Germany postal zones	0.29451×10^{12}	23.40	0.00	7.04	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	78.851	13.35	0.00	0.00	0.00	0.00	0.01	0.00	0.11
4	first Germany postal zones	0.10447×10^{12}	156.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	57.228	11.26	2.40	0.00	0.00	0.00	0.00	0.00	0.16
5	first Germany postal zones	0.59762×10^{11}	315.28	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	46.446	13.83	3.97	1.46	0.00	0.00	0.06	0.00	0.19
6	first Germany postal zones	0.35909×10^{11}	531.44	27.70	11.06	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	39.040	16.21	4.26	0.01	0.00	0.00	0.07	0.00	0.00
7	first Germany postal zones	0.21983×10^{11}	832.60	44.00	7.07	0.00	0.00	0.00	1.48	0.00
	Fisher's iris	34.298	17.43	2.86	0.00	0.00	0.02	1.55	1.34	0.00
8	first Germany postal zones	0.13385×10^{11}	1239.64	0.24	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	29.989	20.81	2.76	0.02	0.00	0.01	0.25	0.00	0.00
9	first Germany postal zones	0.84237×10^{10}	1697.17	28.59	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	27.786	18.80	1.62	0.00	0.00	0.01	0.82	1.37	0.23
10	first Germany postal zones	0.64465×10^{10}	1638.30	0.16	0.00	0.00	0.00	0.00	11.32	0.00
	Fisher's iris	25.834	17.43	2.84	0.42	0.00	0.51	1.00	0.00	0.28

4.2.1. First Clustering Test Problems Situation

The numerical experimental results in Table 4 were taken for seven runs. In these tables, K refers to the number of clusters, and f_{opt} refers to the best identified value.

The proposed MHTSASM algorithm can reach the best results compared with all other clusters techniques for the German towns dataset, as shown in Table 4. The MHTSASM, TS, and GA results were the same. SA and Algorithm 1 gave the same results for all clusters.

Furthermore, the proposed MHTSASM algorithm achieved the best results for all numbers of clusters for the first Bavarian postal zones dataset, as shown in Table 4. The results from Algorithm 1 and MHTSASM were the same. Cluster three gave bad results for K-M, TS, GA, and SA. Cluster four gave better results for K-M, TS, GA, and SA than cluster three. Furthermore, GA gave good results except for cluster three.

The results presented in Table 4 show that TS achieved the best results for all numbers of clusters except for clusters five, where it gave a bad result for the second Bavarian postal zones dataset. The results from Algorithm 1 and the MHTSASM were the same for all clusters. Cluster two gave bad results for all algorithms except TS. Clusters three and four gave good results for the MHTSASM, Algorithm 1, and GA but bad results for SA and K-M. Cluster five gave good results for the proposed MHTSASM algorithm and Algorithm 1 but bad for K-M, TS, GA, and SA.

The results indicated that the proposed MHTSASM algorithm could reach better or very similar solutions to those found using the global optimization methods. Therefore, the proposed MHTSASM algorithm can deeply compute the local minimum in the clustering problem for the objective function.

4.2.2. Second Clustering Test Problems Situation

Another two datasets called (1) the first Germany postal zones dataset, and (2) the Fisher´s iris dataset that includes 150 instances and 4 attributes [8,21] were used in order to evaluate the proposed MHTSASM algorithm with other clustering techniques, such as K-M, J-M+, VNS, VNS+, VNS-1, G1, and G2. The numerical experimental results for this evaluation is presented in Table 5, where K refers to the number of clusters, and f_{opt} refers to the best identified value [28].

The average results for 10 restarts for KM, J-M+, and VNS were used, as seen in Table 5. The results of the first Germany postal zones dataset indicated that the proposed MHTSASM algorithm achieved better results than those for K-M, J-M+, VNS, G1, G2, and Algorithm 1. The proposed MHTSASM algorithm performed the same as VNS-1.

As shown in Table 5, the results of the Fisher´s iris dataset indicated that the proposed MHTSASM algorithm achieved better performance than the K-M technique for all numbers of clusters except for K = 2 and better results than J-M+ for K = 2, 3 [28]. The MHTSASM gave better results than G2 for K = 6, 7, 8, 9, 10. The proposed MHTSASM algorithm and VNS-1 gave results for K = 6, 7, 8 better than all other algorithms. Algorithm 1 and G1 were quite similar; they gave better results than the proposed MHTSASM algorithm for K = 2, 3, 4, 5. VNS gave better results than the MHTSASM for K = 2, 3, 4, 9. Compared with the Algorithm 1, the G1 technique achieved similar or better performance for all K values excluding K = 10. The proposed MHTSASM algorithm gave better results than Algorithm 1 for K = 7, 9. It gave better results than G2 for K = 6, 7, 8, 9, 10. It gave better results than G1 for K = 7, 8, 10. In addition, it gave better results than VNS for K = 6, 8, 10. The VNS-1 can be considered as the best technique for the Fisher´s iris dataset, as it could achieve the best identified optimal values. Hence, the results indicated that the results deviation found using the proposed MHTSASM algorithm from the comprehensive minimum was equal to or less than zero for all K.

4.2.3. Third Clustering Test Problems Situation

Another five datasets, known as Iris, Glass, Cancer, Contraceptive Method Choice (CMC), and Wine were employed to validate the proposed MHTSASM algorithm. Furthermore, there were several methods called (1) KHM, (2) PSO, and (3) PSOKHM algorithm [25] that were selected in order to compare their performances [28]. In addition, methods called (1) KHM, (2) ACO, and (3) ACOKHM algorithm [26,27] were selected for the same purpose.

The clustering methods quality, which could be measured using the F-measure criteria, was compared. F-measure is an external quality measure applicable for quality evaluation tasks [48–52]. It uses the ideas of precision and recall from information retrieval [53–57]. It finds how each point is clustered correctly relative to its original class. Every class (i) assumed using the benchmark dataset class labels is observed as n_i items set chosen for the query. Every cluster (j) created using the data cluster algorithms is observed as n_i items set for the query retrieval. Let n_{ij} denotes the class i element numbers inside cluster j. For every class i inside cluster j, the precision and recall are then defined as shown in Equation (6):

$$P_{(i,j)} = \frac{n_{ij}}{n_j},\ r_{(i,j)} = \frac{n_{ij}}{n_i} \tag{6}$$

and the corresponding value under the F-measure is calculated by Equation (7):

$$F_{(i,j)} = \frac{\left(b^2+1\right) * P_{(i,j)} * r_{(i,j)}}{b^2 * P_{(i,j)} + r_{(i,j)}} \tag{7}$$

where we are assuming that (b = 1) in order to gain equal weighting for $r_{(i,j)}$ and $p_{(i,j)}$. The total F-measure for the dataset of size n can be calculated using Equation (8):

$$F = \sum_i \frac{n_i}{n} \max_j \left\{F_{(i,j)}\right\} \tag{8}$$

The results in Table 6 show the quality of clustering evaluated using the F-measure over five real datasets. The results show standard deviations and means for 10 independent runs.

Table 6. Results of KHM, PSO, PSOKHM, and the proposed MHTSASM algorithm when p = 2.5, 3, and 3.5, respectively.

p No	Dataset Name	KHM	PSO	PSOKHM	MHTSASM
2.5	Iris	0.750	0.711	0.753	0.892
	Glass	0.421	0.387	0.424	0.548
	Cancer	0.829	0.819	0.829	0.872
	CMC	0.335	0.298	0.333	0.404
	Wine	0.516	0.512	0.516	0.715
3	Iris	0.744	0.740	0.744	0.892
	Glass	0.422	0.378	0.427	0.548
	Cancer	0.834	0.817	0.834	0.872
	CMC	0.303	0.250	0.303	0.404
	Wine	0.538	0.519	0.553	0.715
3.5	Iris	0.770	0.660	0.762	0.892
	Glass	0.396	0.373	0.396	0.548
	Cancer	0.832	0.820	0.835	0.872
	CMC	0.332	0.298	0.332	0.404
	Wine	0.502	0.530	0.535	0.715

The results of numerical experiments are presented in Table 6 for KHM, PSO, and PSOKHM when p = 2.5 and indicate that the proposed MHTSASM algorithm performed better than results for KHM, PSO, and PSOKHM for all datasets [28]. Furthermore, the PSOKHM performed better than results for KHM and PSO for datasets Iris and Glass. KHM performed better than results for PSO and PSOKHM for the CMC dataset. PSOKHM and KHM performed the same for datasets Wine and Cancer.

The results of numerical experiments are presented in Table 6 for KHM, PSO, and PSOKHM when p = 3 and show that the proposed MHTSASM algorithm performed better than results for KHM, PSO, and PSOKHM for all datasets. The PSOKHM performed better than results for KHM and PSO for datasets Glass and Wine. The PSOKHM and KHM performed the same for datasets Iris, Cancer, and CMC.

The results of numerical experiments are presented in Table 6 for KHM, PSO, and PSOKHM when p = 3.5 and show that the proposed MHTSASM algorithm performed better than results for KHM, PSO, and PSOKHM for all datasets. PSOKHM performed better than results for KHM and PSO for datasets Cancer and Wine. The KHM performed better than results for PSO and PSOKHM for the Iris dataset. The PSOKHM and KHM performed the same for datasets Glass and CMC.

4.3. Discussion

As shown in Table 4, the proposed MHTSASM algorithm could reach the best results compared with all others cluster techniques for the German towns dataset. Furthermore, it achieved the best results for all numbers of clusters for the first Bavarian postal zones dataset. TS achieved the best results for all numbers of clusters except cluster five, where it gave a bad result for the second Bavarian postal zones dataset. Cluster three gave bad results for K-M [24], TS, GA, and SA [23]. Cluster four gave better results for K-M, TS, GA, and SA than cluster three. Furthermore, GA gave good results except for cluster three. The results indicated that the proposed MHTSASM algorithm could reach better or very similar solutions to those found using the global optimization methods. Therefore, the proposed MHTSASM algorithm could deeply compute the local minimum in the clustering problem for the objective function.

As shown in Table 5, the results of the Fisher´s iris dataset indicated that the proposed MHTSASM algorithm achieved better performance than the K-M [42] technique for all numbers of clusters, except for K = 2, and better results than J-M+ for K = 2, 3. Hence, the results indicate that the results deviation found using the proposed MHTSASM algorithm from the comprehensive minimum is equal to or less than zero for all K.

As shown in Table 6, the results of numerical experiments indicate that the proposed MHTSASM algorithm performed better than results for KHM [25], PSO, and PSOKHM [26,27] for all datasets.

5. Conclusions

Clustering is a common data analysis and data mining problem. It aims to group similar data objects into sets of disjoint classes. Symmetry can be considered as a pre-attentive feature which that improves the shapes and objects, as well as reconstruction and recognition. The symmetry-based clustering methods search for clusters that are symmetric with respect to their centers. In addition, the K-M algorithm is one of the most common clustering methods. It can be easily implemented and works faster in most conditions. However, it is sensitively initialized and it can be easily trapped in the local targets. The TS algorithm is a stochastic global optimization technique. A new algorithm using the TS with K-M clustering, called MHTSASM, was presented in this paper. The MHTSASM algorithm fully uses the merits of both TS and K-M algorithms. It uses TS to make economic exploration for data with the help of ASM. It uses different strategies of TS, such as the intensification and diversification. The proposed MHTSASM algorithm performance is compared with multiple clustering techniques based on both optimization and meta-heuristics. The experimental results ensure that the MHTSASM overcomes initialization sensitivity of K-M and reaches the global optimal effectively. However, dealing and identifying clusters with non-convex shapes is one of the paper's limitation, and it will be one of the future research direction.

Funding: This research is funded by Deanship of Scientific Research at Umm Al-Qura University, Grant Code: 22UQU4281768DSR02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Acknowledgments: The author would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4281768DSR02).

Conflicts of Interest: The author declares no conflict of interest.

References

1. Bharany, S.; Sharma, S.; Badotra, S.; Khalaf, O.I.; Alotaibi, Y.; Alghamdi, S.; Alassery, F. Energy-Efficient Clustering Scheme for Flying Ad-Hoc Networks Using an Optimized LEACH Protocol. *Energies* **2021**, *14*, 6016. [CrossRef]
2. Li, G.; Liu, F.; Sharma, A.; Khalaf, O.I.; Alotaibi, Y.; Alsufyani, A.; Alghamdi, S. Research on the natural language recognition method based on cluster analysis using neural network. *Math. Probl. Eng.* **2021**, *2021*, 9982305. [CrossRef]
3. Jinchao, J.; Li, W.P.Z.; He, F.; Feng, G.; Zhao, X. Clustering Mixed Numeric and Categorical Data with Cuckoo Search. *IEEE Access* **2020**, *8*, 30988–31003.
4. Su, M.C.; Chou, C.H. A modified version of the K-means algorithm with a distance based on cluster symmetry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 674–680. [CrossRef]
5. Vijendra, S.; Laxman, S. Symmetry based automatic evolution of clusters: A new approach to data clustering. *Comput. Intell. Neurosci.* **2015**, *2015*, 796276. [CrossRef] [PubMed]
6. Khan, S.S.; Ahmad, A. Cluster center initialization algorithm for K-means clustering. *Pattern Recognit. Lett.* **2004**, *25*, 1293–1302. [CrossRef]
7. Alotaibi, Y.; Subahi, A.F. New goal-oriented requirements extraction framework for e-health services: A case study of diagnostic testing during the COVID-19 outbreak. *Bus. Process Manag. J.* **2021**, *28*. [CrossRef]
8. Subramani, N.; Mohan, P.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. An Efficient Metaheuristic-Based Clustering with Routing Protocol for Underwater Wireless Sensor Networks. *Sensors* **2022**, *22*, 415. [CrossRef] [PubMed]
9. Huizhen, Z.; Liu, F.; Zhou, Y.; Zhang, Z. A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem. *Inf. Sci.* **2020**, *539*, 347–374.
10. Suganthi, S.; Umapathi, N.; Mahdal, M.; Ramachandran, M. Multi Swarm Optimization Based Clustering with Tabu Search in Wireless Sensor Network. *Sensors* **2022**, *22*, 1736. [CrossRef] [PubMed]
11. Kareem, S.S.; Mostafa, R.R.; Hashim, F.A.; El-Bakry, H.M. An Effective Feature Selection Model Using Hybrid Metaheuristic Algorithms for IoT Intrusion Detection. *Sensors* **2022**, *22*, 1396. [CrossRef] [PubMed]
12. Rajendran, S.; Khalaf, O.I.; Alotaibi, Y.; Alghamdi, S. MapReduce-based big data classification model using feature subset selection and hyperparameter tuned deep belief network. *Sci. Rep.* **2021**, *11*, 24138.
13. Liqin, Y.; Cao, F.; Gao, X.Z.; Liu, J.; Liang, J. k-Mnv-Rep: A k-type clustering algorithm for matrix-object data. *Inf. Sci.* **2020**, *542*, 40–57.
14. Mansouri, N.; Javidi, M.M. A review of data replication based on meta-heuristics approach in cloud computing and data grid. *Soft Comput.* **2020**, *24*, 14503–14530. [CrossRef]
15. Li, J.Q.; Duan, P.; Cao, J.; Lin, X.P.; Han, Y.Y. A hybrid Pareto-based tabu search for the distributed flexible job shop scheduling problem with E/T criteria. *IEEE Access* **2018**, *6*, 58883–58897. [CrossRef]
16. Amir, A.; Khan, S.S. Survey of state-of-the-art mixed data clustering algorithms. *IEEE Access* **2019**, *7*, 31883–31902.
17. Laith, A. Group search optimizer: A nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Comput. Appl.* **2020**, *33*, 2949–2972.
18. Alotaibi, Y. A New Database Intrusion Detection Approach Based on Hybrid Meta-Heuristics. *CMC-Comput. Mater. Contin.* **2021**, *66*, 1879–1895. [CrossRef]
19. Alotaibi, Y.; Malik, M.N.; Khan, H.H.; Batool, A.; ul Islam, S.; Alsufyani, A.; Alghamdi, S. Suggestion Mining from Opinionated Text of Big Social Media Data. *CMC-Comput. Mater. Contin.* **2021**, *68*, 3323–3338. [CrossRef]
20. Kosztyán, Z.T.; Telcs, A.; Abonyi, J. A multi-block clustering algorithm for high dimensional binarized sparse data. *Expert Syst. Appl.* **2022**, *191*, 116219. [CrossRef]
21. Chou, X.; Gambardella, L.M.; Montemanni, R. A tabu search algorithm for the probabilistic orienteering problem. *Comput. Oper. Res.* **2021**, *126*, 105107. [CrossRef]
22. Yiyong, X.; Huang, C.; Huang, J.; Kaku, I.; Xu, Y. Optimal mathematical programming and variable neighborhood search for k-modes categorical data clustering. *Pattern Recognit.* **2019**, *90*, 183–195.
23. Amit, S.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O.P.; Tiwari, A.; Er, M.J.; Ding, W.; Lin, C.T. A review of clustering techniques and developments. *Neurocomputing* **2017**, *267*, 664–681.
24. Rabbani, M.; Mokhtarzadeh, M.; Manavizadeh, N. A constraint programming approach and a hybrid of genetic and K-means algorithms to solve the p-hub location-allocation problems. *Int. J. Manag. Sci. Eng. Manag.* **2021**, *16*, 123–133. [CrossRef]
25. Ali, N.A.; Han, F.; Ling, Q.H.; Mehta, S. An improved hybrid method combining gravitational search algorithm with dynamic multi swarm particle swarm optimization. *IEEE Access* **2019**, *7*, 50388–50399.
26. Amir, A.; Hashmi, S. K-Harmonic means type clustering algorithm for mixed datasets. *Appl. Soft Comput.* **2016**, *48*, 39–49.
27. Sina, K.; Adibeig, N.; Shanehbandy, S. An improved overlapping k-means clustering method for medical applications. *Expert Syst. Appl.* **2017**, *67*, 12–18.
28. Adil, B.M.; Yearwood, J. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. *Eur. J. Oper. Res.* **2006**, *170*, 578–596.
29. Mustafi, D.; Sahoo, G. A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the k-means algorithm with applications in text clustering. *Soft Comput.* **2019**, *23*, 6361–6378. [CrossRef]
30. Rout, R.; Parida, P.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. Skin Lesion Extraction Using Multiscale Morphological Local Variance Reconstruction Based Watershed Transform and Fast Fuzzy C-Means Clustering. *Symmetry* **2021**, *13*, 2085. [CrossRef]

31. Mohan, P.; Subramani, N.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I.; Ulaganathan, S. Improved Metaheuristics-Based Clustering with Multihop Routing Protocol for Underwater Wireless Sensor Networks. *Sensors* **2022**, *22*, 1618. [CrossRef] [PubMed]
32. Yan, Y.; He, M.; Song, L. Evaluation of Regional Industrial Cluster Innovation Capability Based on Particle Swarm Clustering Algorithm and Multi-Objective Optimization. Available online: https://www.springerprofessional.de/en/evaluation-of-regional-industrial-cluster-innovation-capability-/19688906 (accessed on 20 January 2022).
33. Chen, L.; Guo, Q.; Liu, Z.; Zhang, S.; Zhang, H. Enhanced synchronization-inspired clustering for high-dimensional data. *Complex Intell. Syst.* **2021**, *7*, 203–223. [CrossRef]
34. Sung-Soo, K.; Baek, J.Y.; Kang, B.S. Hybrid simulated annealing for data clustering. *J. Soc. Korea Ind. Syst. Eng.* **2017**, *40*, 92–98.
35. Yi-Tung, K.; Zahara, E.; Kao, I.W. A hybridized approach to data clustering. *Expert Syst. Appl.* **2008**, *34*, 1754–1762.
36. Ezugwu, A.E.; Adeleke, O.J.; Akinyelu, A.A.; Viriri, S. A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. *Neural Comput. Appl.* **2020**, *32*, 6207–6251. [CrossRef]
37. Alsufyani, A.; Alotaibi, Y.; Almagrabi, A.O.; Alghamdi, S.A.; Alsufyani, N. Optimized Intelligent Data Management Framework for a Cyber-Physical System for Computational Applications. Available online: https://link.springer.com/article/10.1007/s40747-021-00511-w (accessed on 1 January 2022).
38. Pasi, F.; Sieranoja, S. How much can k-means be improved by using better initialization and repeats? *Pattern Recognit.* **2019**, *93*, 95–112.
39. Xiao-Dong, W.; Chen, R.C.; Yan, F.; Zeng, Z.Q.; Hong, C.Q. Fast adaptive K-means subspace clustering for high-dimensional data. *IEEE Access* **2019**, *7*, 42639–42651.
40. Asgarali, B.; Hatamlou, A. An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms. *Appl. Soft Comput.* **2018**, *67*, 172–182.
41. Yinhao, L.; Cao, B.; Rego, C.; Glover, F. A Tabu Search based clustering algorithm and its parallel implementation on Spark. *Appl. Soft Comput.* **2018**, *63*, 97–109.
42. Alotaibi, Y. A New Secured E-Government Efficiency Model for Sustainable Services Provision. *J. Inf. Secur. Cybercrimes Res.* **2020**, *3*, 75–96. [CrossRef]
43. Costa, L.R.; Aloise, D.; Mladenović, N. Less is more: Basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering. *Inf. Sci.* **2017**, *415*, 247–253. [CrossRef]
44. Gribel, D.; Vidal, T. HG-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering. *Pattern Recognit.* **2019**, *88*, 569–583. [CrossRef]
45. Safwan, C.F. A Genetic Algorithm that Exchanges Neighboring Centers for Fuzzy c-Means Clustering. Ph.D. Thesis, Nova Southeastern University, Lauderdale, FL, USA, 2012.
46. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [CrossRef]
47. Alotaibi, Y. Automated Business Process Modelling for Analyzing Sustainable System Requirements Engineering. In Proceedings of the 2020 6th International Conference on Information Management (ICIM), London, UK, 27–29 March 2020; pp. 157–161.
48. Katebi, J.; Shoaei-parchin, M.; Shariati, M.; Trung, N.T.; Khorami, M. Developed comparative analysis of metaheuristic optimization algorithms for optimal active control of structures. *Eng. Comput.* **2020**, *36*, 1539–1558. [CrossRef]
49. Majid, E.; Shahmoradi, H.; Nemati, F. A new preference disaggregation method for clustering problem: DISclustering. *Soft Comput.* **2020**, *24*, 4483–4503.
50. Rawat, S.S.; Alghamdi, S.; Kumar, G.; Alotaibi, Y.; Khalaf, O.I.; Verma, L.P. Infrared Small Target Detection Based on Partial Sum Minimization and Total Variation. *Mathematics* **2022**, *10*, 671. [CrossRef]
51. Ashraf, F.B.; Matin, A.; Shafi, M.S.R.; Islam, M.U. An Improved K-means Clustering Algorithm for Multi-dimensional Multi-cluster data Using Meta-heuristics. In Proceedings of the 2021 24th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 18–20 December 2021; pp. 1–6.
52. Zhao, L.; Wang, Z.; Zuo, Y.; Hu, D. Comprehensive Evaluation Method of Ethnic Costume Color Based on K-Means Clustering Method. *Symmetry* **2021**, *13*, 1822. [CrossRef]
53. Beldjilali, B.; Benadda, B.; Sadouni, Z. Vehicles Circuits Optimization by Combining GPS/GSM Information with Metaheuristic Algorithms. Rom. *J. Inf. Sci. Technol.* **2020**, *23*, T5–T17.
54. Zamfirache, I.A.; Precup, R.E.; Roman, R.C.; Petriu, E.M. Reinforcement Learning-based control using Q-learning and gravitational search algorithm with experimental validation on a nonlinear servo system. *Inf. Sci.* **2022**, *583*, 99–120. [CrossRef]
55. Pozna, C.; Precup, R.E.; Horvath, E.; Petriu, E.M. Hybrid Particle Filter-Particle Swarm Optimization Algorithm and Application to Fuzzy Controlled Servo Systems. *IEEE Trans. Fuzzy Syst.* **2022**, *1*. [CrossRef]
56. Thakur, N.; Han, C.Y. A Study of Fall Detection in Assisted Living: Identifying and Improving the Optimal Machine Learning Method. *J. Sens. Actuator Netw.* **2021**, *10*, 39. [CrossRef]
57. Jayapradha, J.; Prakash, M.; Alotaibi, Y.; Khalaf, O.I.; Alghamdi, S. Heap Bucketization Anonymity-An Efficient Privacy-Preserving Data Publishing Model for Multiple Sensitive Attributes. *IEEE Access* **2022**, *1*. [CrossRef]

symmetry

MDPI

Article

A Lévy Distribution Based Searching Scheme for the Discrete Targets in Vast Region

Zhigang Lian [1], Dan Luo [2,*], Bingrong Dai [3] and Yangquan Chen [4]

[1] School of Electronic and Information Engineering, Shanghai Dianji University, Shanghai 201306, China; lianzg@sdju.edu.cn
[2] School of Mathematics and Statistics, Baise University, Baise 533000, China
[3] School of Information Engineering, Shanghai Maritime University, Shanghai 201306, China; brdai@shmtu.edu.cn
[4] School of Engineering, University of California, Merced, CA 95343, USA; chen53@ucmerced.edu
* Correspondence: bsxyldan@163.com

Abstract: This paper investigates the *Discrete Targets Search Problem, (DTSP)*, which aims to quickly search for discrete objects scattered in a vast symmetry region. Different from continuous function extremal value search, the discrete points search cannot make use of the properties of regular functions, such as function analytic, single/multiple extreme, and monotonicity. Thus, in this paper a new search scheme based on Lévy random distribution is investigated. In comparison with the *Traditional Carpet* search or *Random* search based on other distributions, *DTSP* can provide much faster search speed which is demonstrated by simulation with different scales problems for the selected scenarios. The simulations experiment proves that *DTSP* is faster for searching for a discrete single target or multiple targets in a wide area. It provides a new method for solving the discrete target search problem.

Keywords: discrete targets search; grid search/traditional carpet search; uniform distribution search; *Lévy distribution* search; targets search optimization; vast symmetry region

Citation: Lian, Z.; Luo, D.; Dai, B.; Chen, Y. A Lévy Distribution Based Searching Scheme for the Discrete Targets in Vast Region. *Symmetry* **2022**, *14*, 272. https://doi.org/10.3390/sym14020272

Academic Editor: Alexander Shelupanov

Received: 7 November 2021
Accepted: 12 December 2021
Published: 29 January 2022

1. Introduction

This paper is motivated by a world-wide focused case, the lost connection of the Malaysia Airlines Flight 370 (MH370). It has been missing up to now since then. Dozens of countries around the world have attempted the search using different kinds of most advanced technology such as radio "pings" between the aircraft and an Inmarsat satellite based on military radar data. Unfortunately, there has been neither confirmation of flight debris nor indication of crash sites. This paper defines the sea target search, grassland target search, desert target search, etc., in a large area as *Discrete Targets Search Problem* which is *DTSP* for short. *DTSP* has a wide range of applications, such as in the case of war, how to quickly search the enemy warships group. In addition, the rescue of airplane crashes, shipwrecks, and the desert search, etc. can be conversed with *DTSP* to be solved. Because search goal area is very large, the search target is a decentralized point, and characteristics of targets are not clear, *DTSP* is very difficult, and the research report on it is more visible. In a large area, see in Figure 1 for how to quickly search target. Research *DTSP* has great practical meaning.

There are some reports on the study of target recognition search, mainly through radar, infrared, robot, and other tracking. Some pioneer researchers reported about target recognition, target acquisition, and target tracking. For example, references [1,2] considered the case of heuristic search where the location of the goal may change during the course of the search. They introduced ideas from the area of resource-bounded planning into a Moving Target Search (*MTS*) algorithm, including commitment to god and deliberation for selecting plans. J. R. R. Uijlings et al. [3] addresses the problem of generating possible object locations for the use in object recognition. They introduce selective search by combining the strength of both an exhaustive search and segmentation. Grimaldo Silva et al. [4] proposed the use of saliency information to organize regions based on their probability of containing objects. Xin, ZH et al. introduced a new ground moving-target-focused method for airborne synthetic aperture radar in [5]. Compared to conventional SAR imaging methods, it can eliminate the effects of Doppler ambiguity and azimuth spectrum split. Mohamed Abd Allah El-Hadidy in [6] presented a search model that finds n-dimensional randomly moving target in which no information of the target position is available to the searchers. Zhen Shi et al. in [7] proposed a parallel search strategy based on kernel sparse representation to perform the tracking task in the FLIR sequences. The kernel method is used to deal with complex appearance variations. A variety of different target search problems were reported with different technical strategies, such as $UAVs$ [8] and robots [9–11]. However, there are a few references about the discrete point object search problem. This paper discusses different strategies to study how to search discrete point object efficiently. In addition, there are more researches on the problem of discrete target search from the aspects of infrared and radar technology but fewer research reports from the perspective of image processing and algorithms.

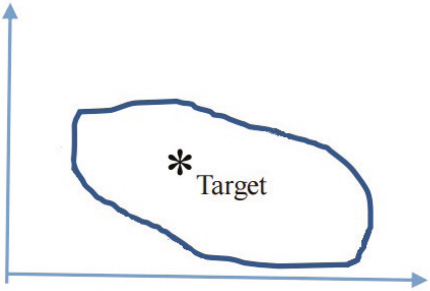


Figure 1. Discrete point target search. * is the label of the search target location.

The discrete point search problem is very different from the extreme value of function search problem (see Figure 2). In this case, the objects are isolated, and they are not related to each other. However, in the extreme value of function search problem, extreme points and other points are related, such that intelligent iterative calculation can be used to find their extreme point. Aiming at the particularity of discrete target search problem, this article proposes an efficient and reliable search method to achieve rapid searching target in a wide range of areas. We study the effectiveness of *Traditional Carpet* search, random search, and *Lévy distribution* random search for discrete point object, with the following differences from existing ones: Firstly, based on the practical problems of searching for discrete targets in people's production and life, $DTSP$ is proposed. From an actual target search problem analysis, it summarized and extracted a common technical problem. From the actual target search problem analysis, it summed up the common technical solutions to the problem; secondly, it formulates a new search with different evolution mode, for discrete point object problem; thirdly, it strictly analyzes and compares different search method performance. We show that the proposed search method is more efficient than tradition search and discuss the implication of the results and suggestion for further research.

The remainder of this paper is organized as follows: In Section 2, we sum up the search method for discrete targets in the current reality. In Section 3, we develop the general framework of the search solution. In Section 4, we formulate some dominance properties and model mathematical of $DTSP$. In Section 5, we discuss the *Traditional Carpet* search, random search, and *Lévy distribution* random search for discrete point object. The *Lévy distribution* random search was developed to enhance the search efficiency for the $DTSP$. The results of a computational experiment are provided in Section 6. Some conclusions are given in the last section.

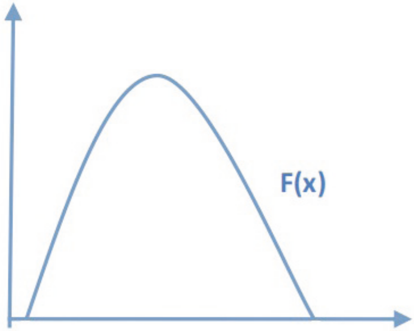


Figure 2. Extreme value of function search.

2. The Current Search Method

The traditional search methods are usually used for search in plane crash, ship sank, and desert missing, as well as the search of landslide spot, forest fire spot and rescue, grassland, and enemy warships group, etc. For example, sending aircraft flights to the target area to search for suspicious targets or sending ground and rescue personnel for on-site search. Most of the current searches are *Traditional Carpet* search, which divide the target area into many lattices to be searched one by one in turn, as shown by Figure 3:

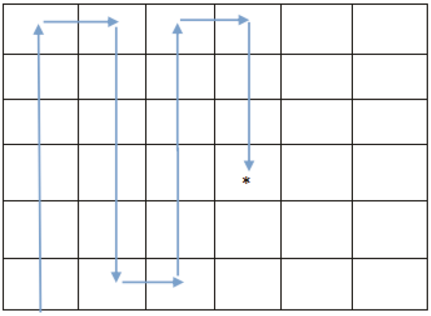


Figure 3. The carpet search model. * is the label of the search target location.

In the past, most actual discrete point object search problems use the *Traditional Carpet* search to look for target. Rescuers use any possible means; aircraft and ships deploy equipment to listen for signals from the underwater locator beacons attached to the aircraft's "black box" flight recorders. However the batteries of the beacons become exhausted in a short time. For example in MH370 search, the search operation continued in the area of the signals using a robotic submarine for a long time but without obtaining any evidence of debris.

The efficiency of *Traditional Carpet* search method is low due to its enumeration nature. It is almost invalid when the target is in a big area. The paper [12] presented a FogLight approach for searching metabolic networks utilizing Boolean (AND-OR) operations represented in matrix notation to efficiently reduce the search space. This paper presents a new idea and method for discrete point object search.

3. New Search Solution Based on Satellite Image

Currently the image processing technology is very mature, and satellite images can be used to look for targets. Methods for image processing include the so-called "M-estimators" (from robust statistics) [13], among others machine learning techniques. This reinforces the credibility of satellite images than can be used in discrete search problems. Robust M-estimators, which are generalizations of maximum-likelihood estimators, tend to look at the bulk of the data/image information and ignore atypical values (outliers or multiple outliers) during estimation procedure (pattern recognition), due to their mathematical structure.

The new algorithm based on satellite images processing operations is as follows:

(1) The search command center informs the satellite control station to adjust satellite attitude and direction to target search area;

(2) The satellite takes a picture of search target area and then promptly sends the image to the processing center;

(3) The image processing center first amplifies satellite photographs and cuts them into the lattice images. Secondly, the algorithm chooses the lattice images to identify which image is the target. Thirdly, it determines the actual position of the target image and then quickly identifies the suspected target coordinates;

(4) It sends the coordinates of the suspected target to the UAV, search aircraft, ship, or ground rescuers to affirm the target and obtain more detailed information;

(5) All information of targets is sent to the search center.

The Discrete Targets search diagram based on the new algorithm with satellite image processing is illustrated in Figure 4:

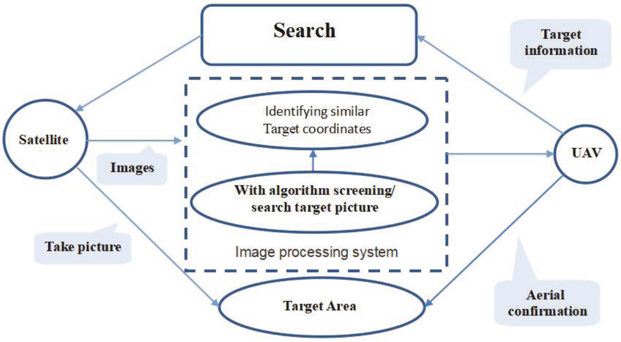


Figure 4. The search logic diagram.

4. Discrete Target Search Problem

4.1. Target (Black Box) Search Problem

In extreme value of function search problem, the extreme point and other points are related and can share useful information for looking for object. However, with the discrete point target search problem, its target is isolated, and all points are not relative each other. With $DTSP$, other points are not able to provide the useful information to help find the target point. For the special case of $DTSP$, it is impossible to use a traditional method such as intelligent iterative calculation to solve it. Here, we present a new method to solve $DTSP$.

Put the target into the coordinate system. If the coordinate value of target is focused, it is a successful search. The problem of discrete point search is translated into a search problem of a target element in the matrix to look for the coordinate of 1 in one target search matrix (1).

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \tag{1}$$

Code rule/0-1: The target area image is divided into lattice images. If a grid image is an object, the element of the grid image is set to 1 in the matrix. Put all lattice images into matrix with the target image denoted by truth/1 and others by false/0. In the search process, if the coordinate of the 1 point is found the search is successful.

4.2. Multi-Targets (Black Box and Wreckage) Search

Put the main target(black box) and wreckage into the coordinate system. If the main target and some wreckage coordinate value are found, it is a successful search. The problem of some discrete target search, is translated into a search problem of target elements, i.e., the coordinates of targets B(black box) and w_i ($0 < w_i < A$ wreckage), in multi-targets matrix (2), in which the B and w_i denote the black box and wreckage, respectively.

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & w_1 & 0 & w_2 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & B & 0 & w_i & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

Code rule/0-*B*-w_i: Put all lattice images into a matrix. Main target/black box image are denoted by B, wreckage is w_i, and others are *false/0*. In matrix (2), B is the goal/black box, w_i is the wreckage, and 0 is blank. A successful search aims to find the black box image or most of wreckage coordinates or the black box and a small part of the wreckage coordinates. If the coordinates of black box B point or wreckage w_i point with the total number more than X are found, the search is successful.

In the following section, we further discuss how to use minimum step to look for the coordinate of targets, how image processing center uses the least number processing picture, and how to judge and look for a target and determine its coordinates.

5. Screening/Search Target Picture

5.1. A Uniform Distribution Search Method

A uniform random search target image method is as follows:

Step 1: Based on $Uniform$ distribution, randomly generate two integers (x, y) as the ordinate and abscissa of the candidate target;

Step 2: It is judged whether or not the generated random number (x, y) is within the search area, and if it is not, using $Uniform$ distribution, reproduce the number of random integers (a, b) as coordinates;

Step 3: Take the satellite grid image in the (x, y) coordinates, judge whether is it real target. If it is not a target, go to step 1.

Step 4: If the target is searched, the coordinates of the target and its image are output;

Above using uniform distribution, the search random extracts the satellite picture judgment target, which is the $Uniform$ distribution search method. Of course, one can also use other random distribution methods to generate coordinates to extract the satellite picture judgment search target.

5.2. A Lévy Distribution Search Method

Lévy stable distributions are a rich class of probability distributions that have many intriguing mathematical properties. *Lévy* flights are a class of random walks in which the step lengths are drawn from a probability distribution with a power law tail. The application of *Lévy* fight pattern in the *Cuckoo* search algorithm [14] greatly improves the algorithm search speed to solve the design optimization [15], selected engineering applications of gradient free optimization [16], traveling salesman problem [17,18], and the selection of optimal machine parameters in milling operations [19]. Tarik Ljouad et al. [20] presented a new tracking approach. Gurjit S. W and Rajiv K [21] proposed an evolutionary particle filter based on improved cuckoo search which overcomes the sample impoverishment problem of generic particle filter. *Lévy* flights are commonly used in physics to model a variety of processes including diffusion. Broadly speaking, *Lévy* flights are a random walk by step size following *Lévy* distribution, and the walking direction is *Uniform* distribution.

In the *Mantegna* rule, steps size s design is as follows:

$$s = \frac{\mu}{|v|^{\frac{1}{\beta}}} \tag{3}$$

The μ, v follows *Uniform* distribution, i.e.,

$$\mu \sim N(0, \sigma_\mu^2), v \sim N(0, \sigma_v^2) \tag{4}$$

Here,

$$\sigma_v = \{\frac{\Gamma(1+\beta) sin(\frac{\pi\beta}{2})}{\Gamma[(1+\beta)/2]\beta * 2^{(\beta-1)/2}}\}^{\frac{1}{\beta}} \tag{5}$$

$$\sigma_\mu = 1 \tag{6}$$

This paper presents the *Lévy* distribution search method for $DTSP$. Based on *Lévy* distribution with the *Mantegna* rule, it generates the ordinate and abscissa (x, y) of the candidate target, in turn grabbing the picture of coordinate's points (x, y) and then determining whether it is real target. The *Lévy* distribution generation coordinates (x, y) of candidate target must be in the search area. This random extracts satellite picture judgment method, that is the *Lévy* distribution search method.

5.3. Pseudo Code of One Target Search

Step 1: Let parameters; including maximum generation *Endgen*;
Step 2: Iteration process;

- Generate stochastically (use *Uniform* distribution or *Lévy* distribution) search start point coordinate (x_0, y_0);
- Judge the picture of the start point coordinate (x_0, y_0) to see whether it is the target;
- t := 0.

While ($t < Endgen$) or (Other stop criterion)

- t := t + 1;
- Generate next step search point coordinate (x_t, y_t) (use *Uniform* distribution or *Lévy* distribution); If the picture of point coordinate (x_t, y_t) has been searched, skip it and re-search;
- Judge the picture of search point coordinate (x_t, y_t) to see whether it is the black box target; If the picture of point coordinate (x_t, y_t) is the black box target, break.

Step 3: Output last step search point coordinates and its image.

5.4. Pseudo Code of Multi-Targets Search

Step 1: Let parameters; including maximum generation *Endgen*; the total *num* of wreckages searched.
Step 2: Iteration process

- Generate stochastically (use *Uniform* distribution or *Lèvy* distribution) search start point coordinate (x_0, y_0);
- Judge the picture of search start point coordinate (x_0, y_0) to see whether it is the black box target or secondary target;
- Count wreckages found;
- t := 0.

While ($t < Endgen$ and secondary target $\geq$ *num*) or (black box target==1) or (Other stop criterion)

- t := t + 1;

- Generate next step search point coordinate (x_t, y_t) (use *Uniform* distribution or *Lèvy* distribution); If the image of point coordinate (x_t, y_t) has been searched, skip it, and re-research;
- Judge the picture of search point coordinate (x_t, y_t) see whether it is the black box target or secondary target;
- Count wreckage found.

Step 3: Output secondary target number and their coordinates, last step search point coordinate, and their images.

6. Experiment for Target Search Problem

6.1. Test Example

To proof-test the effectiveness and performance of $Traditional\ carpet$ search, $Uniform$ distribution and *Lèvy* distribution search for $DTSP$, a set of 10 test examples with different dimensions was employed to evaluate it. The experimental test examples are as follows:

One target search examples:

E1: [0,10]; [0,10]; Target = [6,5].

E2: [0,50]; [0,50]; Target = [31,29].

E3: [0,100]; [0,100]; Target = [51,55].

E4: [0,1000]; [0,1000]; Target = [509,469].

E5: [0,10,000]; [0,10,000]; Target = [5049,5198].

Multi-targets example:

E6: [0,10]; [0,10]; Main Target = [6,4]. Secondary targets = [5,3;6,5]; WN = 1;

E7: [0,50]; [0,50]; Main Target = [26,27]. Secondary targets = [24,30;31,27;28,25]; WN = 2;

E8: [0,100]; [0,100]; Main Target = [65,44]. Secondary targets = [60,39; 71,46; 74,45; 62,40]; WN = 3;

E9: [0,1000]; [0,1000]; Main Target = [654,598]. Secondary Targets = [514,597;735,525;486, 562;753,556;579,499; 425,786]; WN = 4;

E10: [0,10,000]; [0,10,000]; Main Target = [5149,4809]. Secondary targets = [6514, 5897;7035,5035;4986,5562;6013,5026;5179,5099;4125,4986;6056,5165;7022,5983; 7022,5983]; WN = 5.

6.2. Comparison Test with Different Search Methods

This paper compares the minimum search step number, mean search step number, and standard deviation searched by *Traditional Carpet* search, *Uniform* distribution, and *Lévy* distribution search run 32 times. The experimental parameters and results comparison on each target search are listed in Table 1. In the following table, the *Tradition Carpet* search, *Uniform* distribution, and *Lévy* distribution search are called T, U, and L, respectively.

Remark 1. *In Table 1, Ex is the search example; Abbreviation is the search methods shortened form; Min/Average/Std denote that search target used minimum run step, average step, and step standard deviation in run 32 times;* $e + n$ *mean* $\times 10^n$*. Solution denotes whether to find the target, 1 denotes found, and 0 is not found. Succeed times is the successful search times in run 32 times.* Successful rate *is a successful search rate in 32 run times. The better solutions found in different search methods are illustrated with bold letters, and the best minimum average is shown with an underline, respectively.*

Remark 2. *In* Lévy *distribution search* $\alpha \in [0,2]; \beta \in [-1,1]; \Gamma = X; \delta = \Gamma/2$. *If the* Lévy *distribution generation number is negative, we use its absolute value or re-generate with* $Uniform$ *distribution.*

From Table 1, the *Lévy* distribution search has the highest performance since using it has smaller minimum and arithmetic mean search steps found target compared to solutions obtained by others, and especially for bigger size target search problems, it has better search efficiency. In above test examples, the *Lévy* distribution search can obtain a higher search successful rate. If the target is in the search area center, the *Traditional Carpet* search method using less half steps of search area grid cannot find the target, but using *U* or *L* search method with the probability of more 40% and 60% can find the target. From the above experiments, we can find that scale is larger, the performance of *L* is more outstanding compared with *T* and *U* search. For a huge size search problem experiment, because of the experimental conditions limited, the data storage runs out of memory. In the real environment, because the search range is very large, the *Traditional Carpet* search is difficult for finding the target. From simulation results, we can obtain that the *Lévy* distribution search as being clearly better than *Traditional Carpet* search and *Uniform* distribution search for one target search problem. Especially for large size problem, the *L* search has strong ability.

Table 1. The comparison results of *T*, *U*, and *L* search for one target search.

	Search Methods	Abbreviation	Min/Average/Std	Search Solution	Succeed Times	Successful Rate
E1	Traditional	T	50/50/0	0	0	0
EG = 50 X = [0,10]	Uniform distribution	U	5/37.1563/15.7544	1	19	0.5938
Y = [0,10]	Lévy distribution	L	**3**/33.5938 /16.3408	1	**21**	**0.6563**
E2	Traditional	T	1250/1250/0	0	0	0
EG = 1250 X = [0,50]	Uniform distribution	U	89/994.4688/391.3227	1	14	0.4375
Y = [0,50]	Lévy distribution	L	**56**/1.0228×10^3/359.3115	1	14	0.4375
E3	Traditional	T	5000/5000/0	0	0	0
EG = 5000 X = [0,100]	Uniform distribution	U	1070/4179.5/1162.7	1	15	0.4688
Y = [0,100]	Lévy distribution	L	**292**/3390.3/1724.1	1	**20**	**0.625**
E4	Traditional	T	500,000/500,000/0	0	0	0
EG = 500,000 X = [0,1000]	Uniform distribution	U	**1564**/399,200/169,030	1	12	0.375
Y = [0,1000]	Lévy distribution	L	55,876/126,110/384,780	1	**20**	**0.625**
E5	Traditional	T	50,000,000/50,000,000/0	0	0	0
EG = 50,000,000 X = [0,10,000]	Uniform distribution	U	1,344,005/3.9177×10^7/1.237×10^7	1	19	0.4688
Y = [0,10,000]	Lévy distribution	L	223,230/3.0076×10^7/1.8725×10^7	1	**24**	**0.8438**

Search paths, the convergence and succeed search times distribution figures of most effective of *Traditional Carpet* search, *Uniform* distribution, and *Lévy* distribution search run 32 rounds for five one-target search instances are as in Figure 5. For E4 and E5, the search paths are very complex, drawing a figure that almost cannot be seen clearly; therefore, we do not show their search paths.

From Figure 5, in search target path picture we can see *U* and *T* can quickly find target. The *Traditional Carpet* search method is the enumeration method, gradually slow search, efficiency is very low. From surface view *Traditional Carpet* search method, it can find target with a finite step, but the actual search area is too big, and it cannot use large search steps greatly to search. Within finite steps to search the target, using the random search method is more effective than the *Traditional Carpet* search. In addition, when the search area is huge, the *Uniform* distribution search compared to the *Lévy* distribution search efficiency is obviously much lower. In the following section, we test the multi-targets search using different search methods.

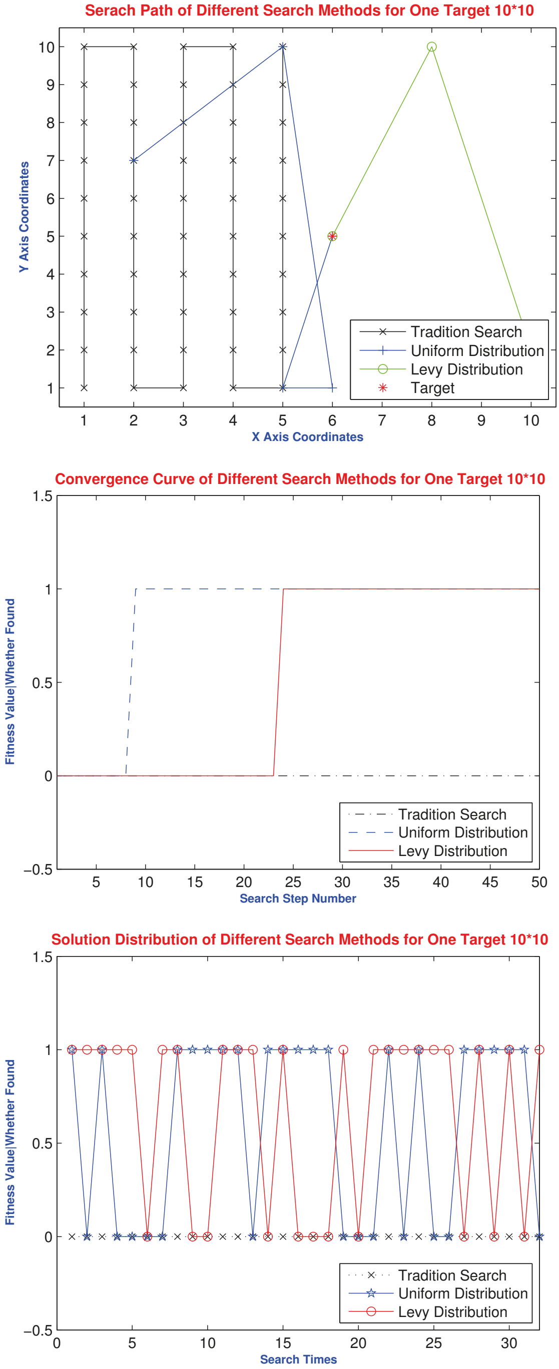


Figure 5. *Cont.*

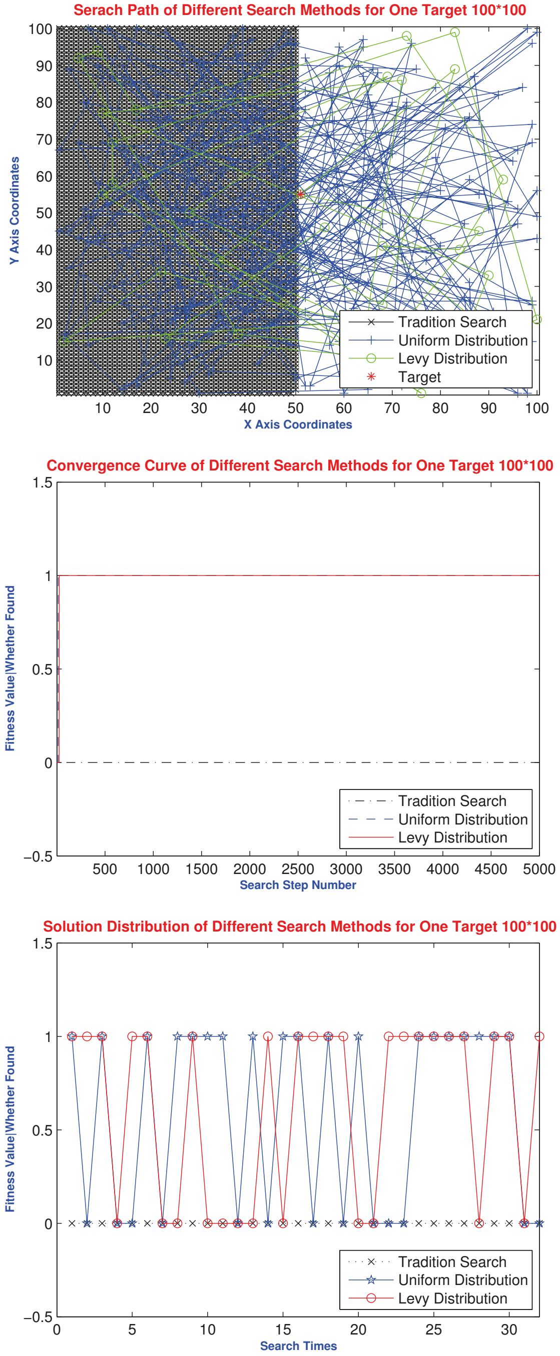


Figure 5. *Cont.*

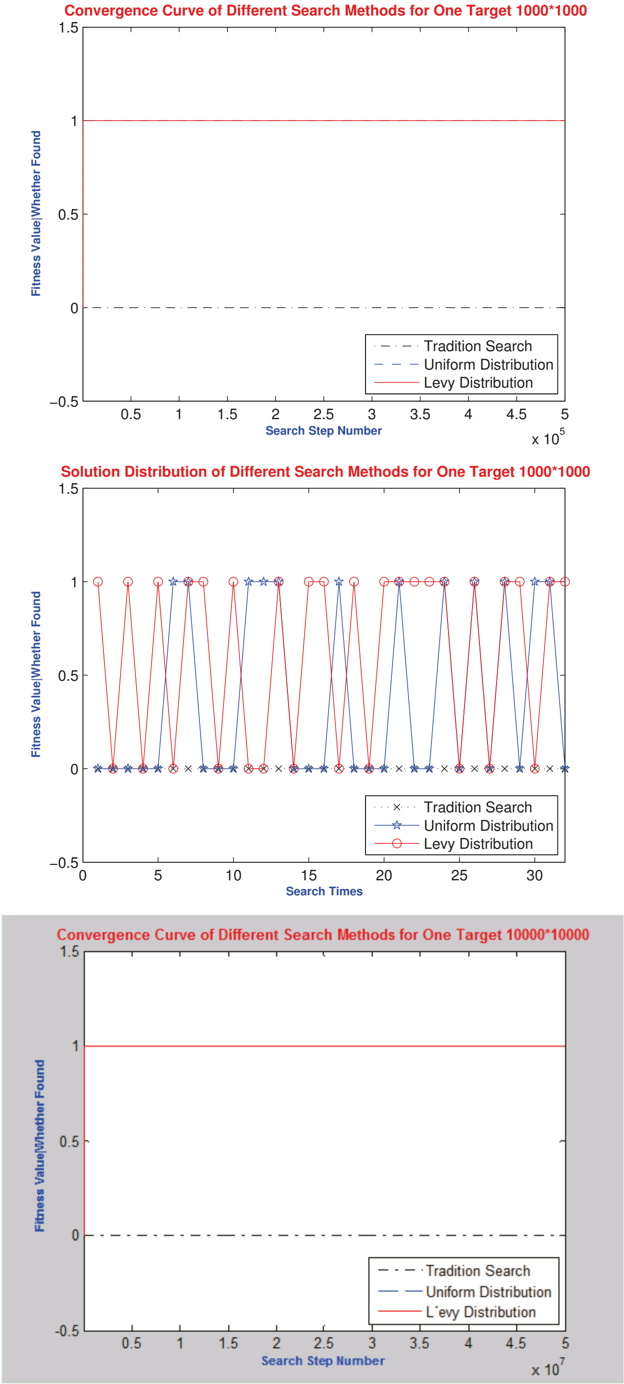


Figure 5. *Cont.*

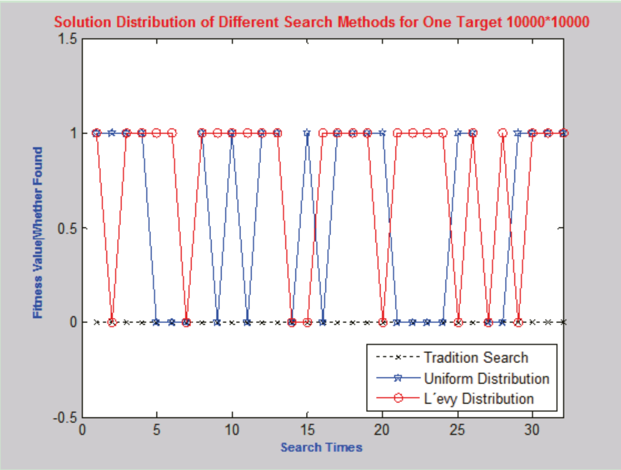


Figure 5. T, U, and L search for one-target.

In order for more scientific evaluation search methods for a multi-targets search problem, this paper compares the minimum search step number, mean search step number, and standard deviation searched by *Traditional carpet* search, *Uniform* distribution, and *Lévy* distribution search run 20 times. The solutions of multi-targets test examples are listed in Table 2, and the about parameters are the same in Table 1.

Table 2. The comparison results of T, U, and L search for multi-targets.

	Search Methods	Min/Average/Std	Main Target			Secondary Targets		
			Search Solution	Succeed Times	Successful Rate	Found Average	Succeed Times	Successful Rate
E6	Traditional	50/50/0	0	0	0	1	20	1
WN = 1 EG = 50	Uniform distribution	29/45.6/7.074	1	9	0.45	0.8	14	0.7
X = [0,10] Y = [0,10]	Lévy distribution	**27**/45.05/7.2655	1	**11**	**0.55**	**1.05**	**18**	**0.9**
E7	Traditional	1250/1250/0	0	0	0	1	0	0
WN = 2 EG = 1250	Uniform distribution	873/1.1872×10^3 129.1191	1	7	0.35	**1.6**	**11**	**0.55**
X = [0,50] Y = [0,50]	Lévy distribution	**753**/1.215×10^3 /117.8882	1	**9**	**0.45**	1.25	7	0.35
E8	Traditional	5000/5000/0	0		0	0	0	0
WN = 3 EG = 5000	Uniform distribution	**2585**/4795 /585.0487	1	8	0.4	1.85	6	0.3
X = [0,100] Y = [0,100]	Lévy distribution	4036/4.8501×10^3 /3339.8648	1	**14**	**0.7**	**2.1**	**7**	**0.35**
E9	Traditional	$5\times 10^5/5 \times 10^5$	0	0	0	2	0	0
WN = 4 EG = 5×10^5	Uniform distribution	355,541/4.8573×10^5 /3.8108×10^4	1	10	0.5	2.9	5	0.25
X = [0,1000] Y = [0,1000]	Lévy distribution	**230,160**/4.4476×10^5 /7.9752×10^4	1	**15**	**0.75**	**3.4**	**12**	**0.6**
E10	Traditional	$2.5 \times 10^6/2.5\times 10^6$ /2.5×10^6	0	0	0	0	0	0
WN = 5 EG = 2.5×10^6	Uniform distribution	$2.5\times 10^6/2.5\times 10^6$ /2.5×10^6	1	5	0.25	2	0	0
X = [0,10,000] Y = [0,10,000]	Lévy distribution	**16,166,752**/2.3883×10^7 /2.7461×10^6	1	**9**	**0.45**	**3.85**	**7**	**0.35**

From Table 2, the *Lévy* distribution search clearly has the highest performance, and especially for bigger size targets search problem, it has better search efficiency. In above test examples, the *Lévy* distribution search can obtain higher main and secondary search successful rate; at the same time, it can find more secondary targets, as well as use less search steps. Especially for large size problems, *L* optimal has strong search ability.

The search paths, the convergence, and succeed search times distribution figures of most effective of $Traditional\ Carpet$ search, $Uniform$ distribution, and *Lévy* distribution search run 20 rounds for five multi-targets search instances are in Figure 6. At the same time, for E8∼E10, the search paths are very complex and cannot be seen clearly; therefore, we do not show their search paths.

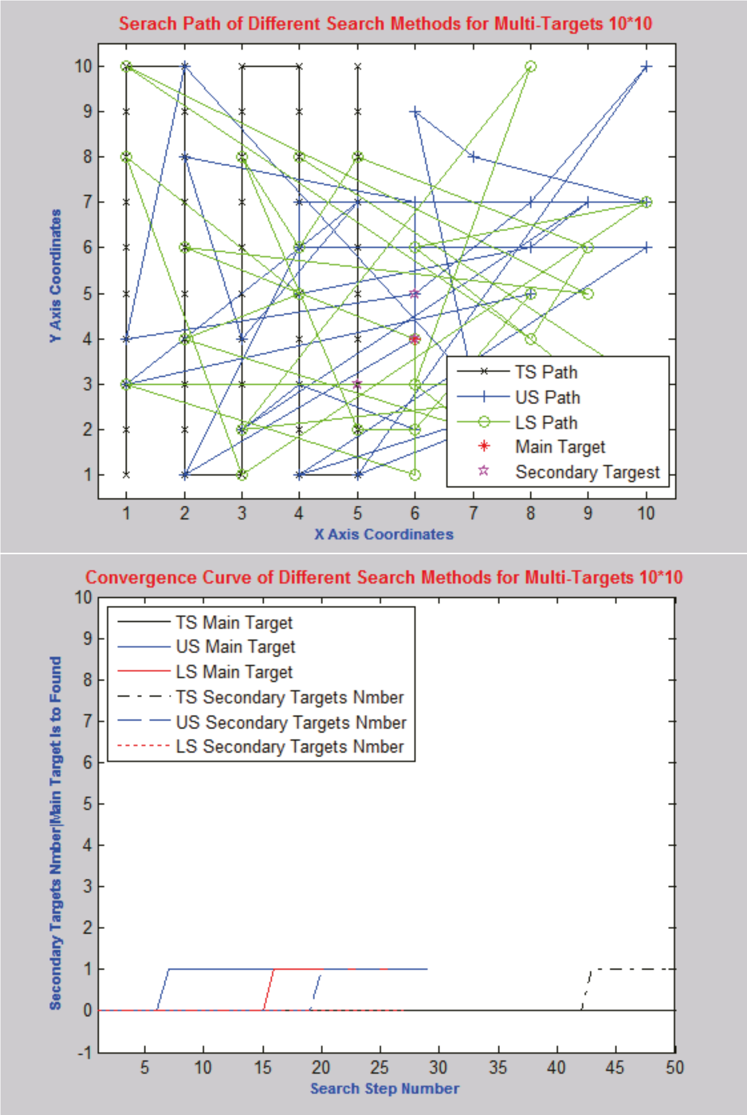


Figure 6. *Cont.*

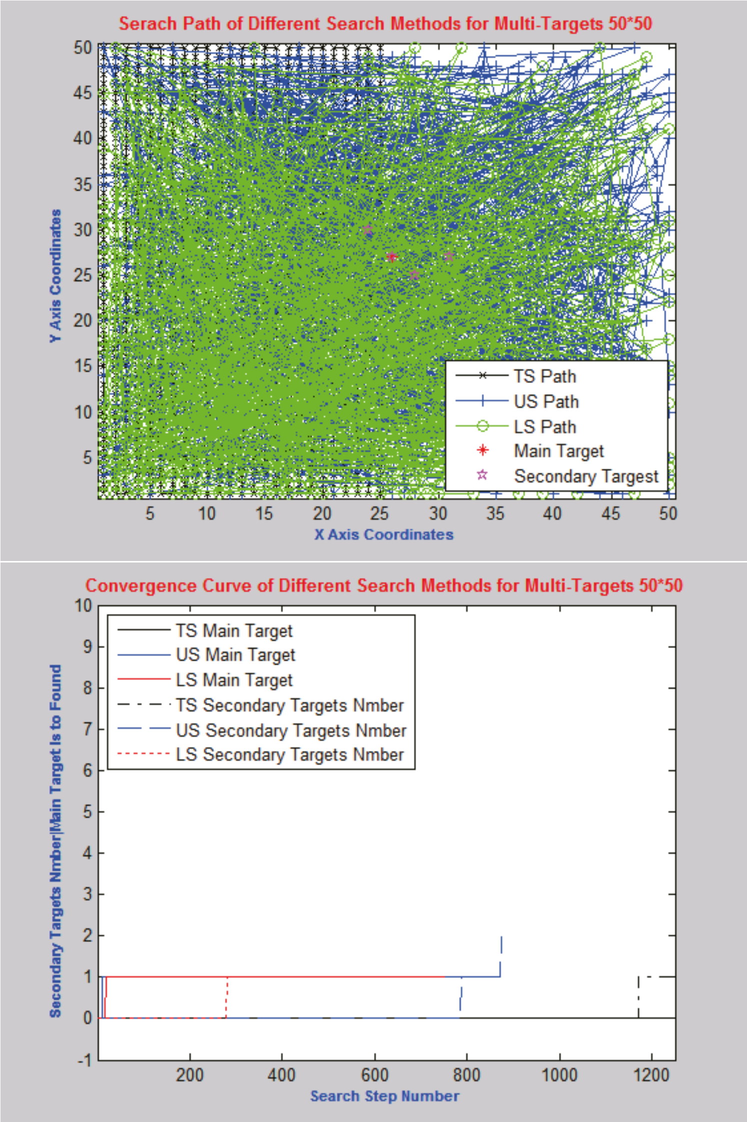


Figure 6. *Cont.*

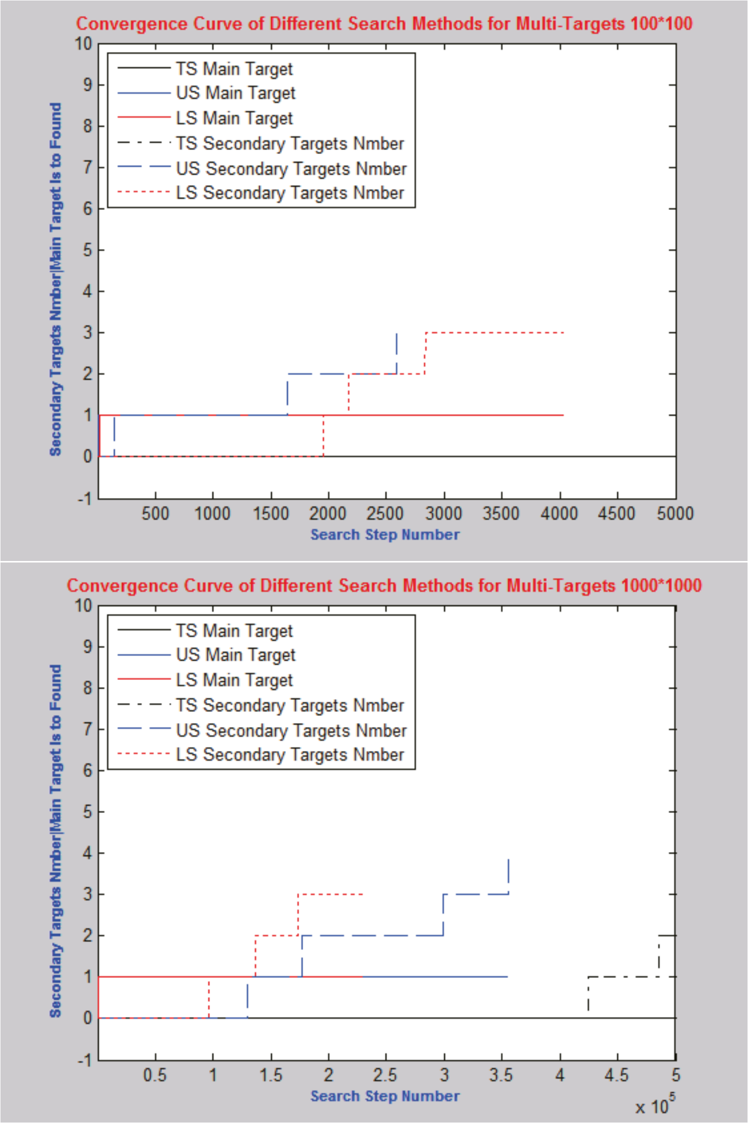


Figure 6. *Cont.*

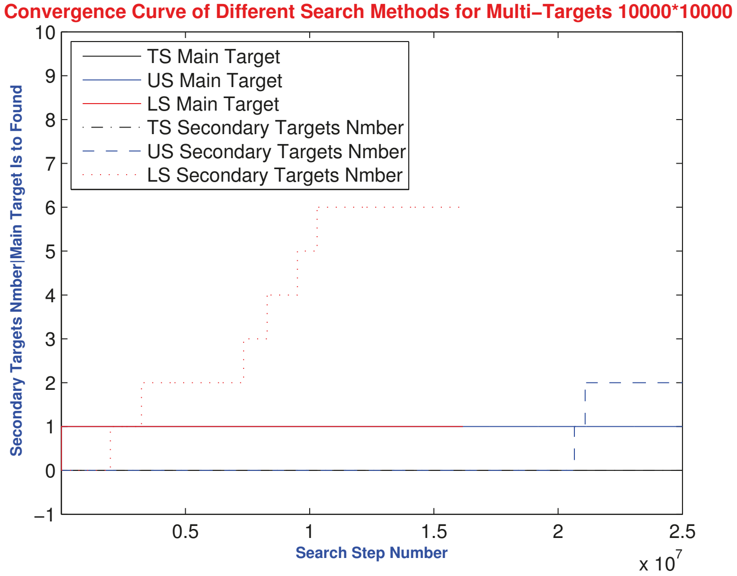


Figure 6. *T*, *U*, and *L* search for multi-targets.

From Figure 6 we can see that the convergence rate of *Lévy* distribution search is clearly faster than *Traditional Carpet* search and *Uniform* distribution search. *Lévy* distribution search can not only fast find the main target but also find more secondary targets than *T* and *U* search method. Due to the particularity of this kind of target search problem, in general secondary targets stochastically scattered on the main target around, it is more suitable for *Lévy* distribution search which can quickly find the targets.

Targets can be found by only processing a few images (search steps) using *Lévy* distribution search. In the scenario under discussion, the advantages lie in that the convergence speed of *Lévy distribution* search being clearly faster than *Traditional Carpet* search and random search based on some other distributions. In limited search steps, *Lévy distribution* search can not only quickly find the main target but also find more secondary targets than the aforementioned search methods.

Accordingly, we can state that the *Lévy* distribution search is more effective than *Traditional Carpet* search and *Uniform* distribution search method for discrete targets search. In limited search steps, the *Lévy* distribution search can, with higher probability, quickly and accurately find the main target and more secondary targets. *Lévy* distribution search is one better search tool.

6.3. Experiment of Lévy Distribution Search

From the above experiment, we find that the *Lévy* distribution search is efficient for discrete targets search. *Lévy* distribution search has many parameters, in order to find their different search performance, we simulate and obtain optimal parameters. *Lévy* distributions solve the test problems, in which α from 0 to 2 with 0.1 step changes and β from -1 to 1 with 0.1 step changes for the X axis and for Y axis, respectively. The L uses a group of parameters α and β combination run 10 times, using the average to search results. The best parameters configuration of L for search one target and multi-targets are in the following Table 3.

Table 3. The optimum parameters of L for target search.

Main Target of One Target Search			Main Target of Multi-Target Search			
Test Problem	**Opimum Average**	α, β	**Test Problem**	**Main Target**	**Secondary Targets**	α, β

E1
EG = 50
	0.9	1.8, −0.9
		0.4, −0.6
		0.2, −0.4
		0.8, −0.3
		1.0, −0.3
		1.2, −0.2
		1.3, −0.1
		1.5, 0.1
		1.4, 0.7
		1.4, 0.8
		0.2, 0.9
			E6
			WN = 1
			EG = 50
				1	1.8	0.1, −0.5;
						0.1, −0.1;
						0.8, 0.

E2
EG = 1250
	0.9	1.2, 0.5	E7
			WN = 2
			EG = 1250
				1	2.5	1.3, −0.9;
						1.3, −0.7;
						1.2, −0.5
						0.9, 0.4;
						0.9, 0.5;
						0.8, 0.9;
						0.8, 1

E3 EG = 5000	1	1.8, −0.5 0.5, 1.0	E8 WN = 3 EG = 5000	1	2.9	0.3, −0.7
E4 EG = 500,000	1	0.4, −0.1 0.9, 0.3 0.5, 0.9	E9 WN = 4 EG = 500,000	1	4.8	0.4, 0.8; 0.2, 0.9

From above Table 3, we can see when β approximately equal to 0.9 have a good search performance. Especially for large-scale one target and multi-targets search problems, β equal to 0.9 is effective and stable. About α for L is not obvious influence and is not stable, relatively speaking in most cases, α is equal to 1.5 has a better search efficiency. As a result of the experiment and paper length limit, this paper do not analyze testing results for large problems E4 and E5.

With the *Lévy* distributions search run 10 times with X and Y each parameters combination, and use the mean main target and the mean secondary targets number of round 10 times, respectively, as Z axis, the three-dimensional graphics are as following (Figures 7 and 8):

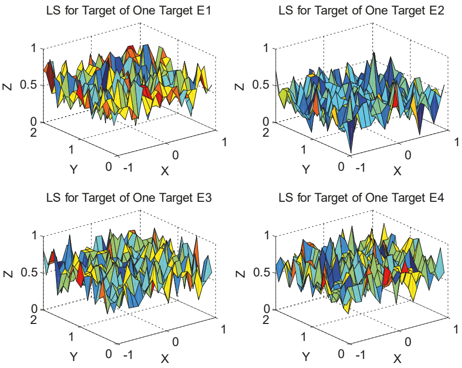


Figure 7. The 3-*D Lévy* distribution random search with different parameters for one target.

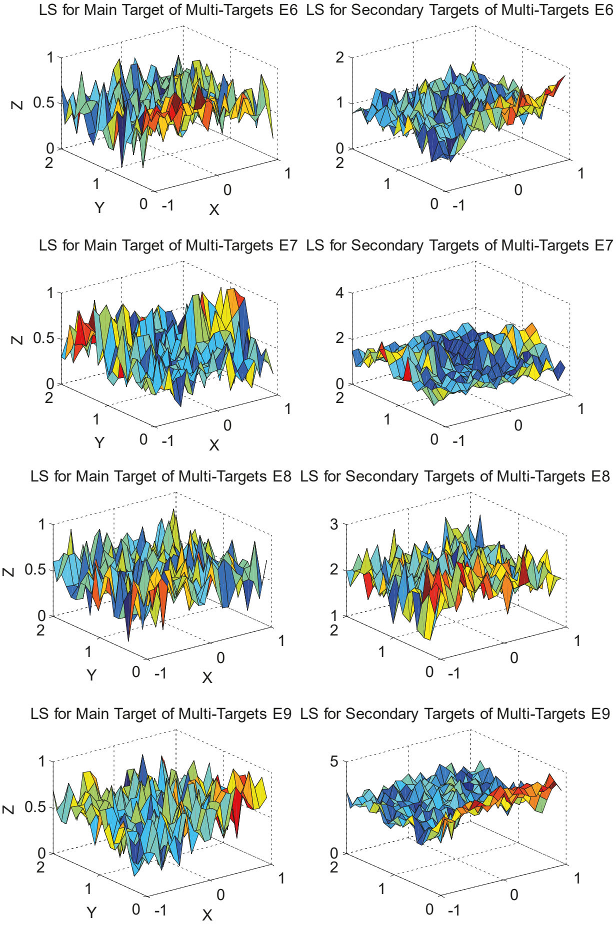


Figure 8. The 3-*D Lévy* distribution random search with different parameters for multi-targets.

From the above figures we can discover that the parameters have greater influence on performance of L algorithm for target search problem. Clearly, we can see when $\beta \approx 0.9$ L has better performance for bigger size target search problem and α influence the performance of L is not stable. Experiments show that L search performance is strong, with fewer steps to find the target.

7. Conclusions and Perspectives

The plane crashed, the ship sank, and desert missing, landslide spot search, forest fire spot search and rescue, grassland search, enemy warships group search, etc., are all called discrete target search problem. According to production and living needs and shortcoming of the tradition search method, this paper offers comprehensive and systematic research on the discrete targets search problem.

This paper analyzes the particularity and complexity of the discrete target search problem, and the paper simulates and analyzes the different and performance characteristics of $Tradition\ Carpet$ search, $Uniform$ distribution search, and *Lévy* distribution search for one target and multi-targets. We give that the computer quickly selects, grabs the satellite images and image analysis method to determine the target coordinate position by software system, and greatly improves the target recognition speed. Preliminary experiments show that the efficiency of *Lévy* distribution search is better than *Tradition Carpet* search and *Uniform* distribution search. Using *Lévy* distribution search only process, a few images (search steps) can find targets. We find the *Lévy* distribution search is one better search tool, which can, with higher probability, quickly and accurately find targets for discrete targets search problem. The simulations experiment prove that *DTSP* is faster to search for discrete single target or multiple targets in a wide area. It provides a new method for solving the discrete target search problem.

There are a number of research directions that can be considered as useful extensions of this research. Although the proposed search method is tested with 10 instances, a more comprehensive computational research should be made to test the efficiency of proposed new method. This paper is the only research on the target of rectangular areas search and study a new method to search target in circular area and irregular graphics area. Recently, 48 robust M-estimators were presented for regression analysis [22]. In addition to image processing, M-estimators have been used in several areas of knowledge such as astronomy, pharmaceutics, medical, econometric, finance, geodesy, and engineering (electrical, telecommunications, civil, mechanical, chemical, and nuclear). In future we will research on other more effective search methods, different distribution search such Mittag-Leffler [23] or multi-population pattern searching algorithm [24] for one target and multi-targets search problem. For future work, as well as in data regression analysis, it is possible to use a function that measures the distance from each location estimated point to the correct locations as performance criteria, principally for multi-targets.

Author Contributions: Conceptualization, Y.C. and Z.L.; Methodology, Y.C. and Z.L.; Investigation, Z.L., D.L. and B.D.; Writing— Z.L., D.L., B.D. and Z.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Shanghai Innovation Project(20AZ10,20AZ11), the Shanghai Educational Science Research Project(Grant No.C2022120), the National Natural Science Foundation of China (Grant No.11661009), the High Level Innovation Teams and Excellent Scholars Program in Guangxi institutions of higher education (Grant No.[2019]52)), the Guangxi Natural Science Key Fund (Grant No.2017GXNSFDA198046), and the Special Funds for Local Science and Technology Development Guided by the Central Government (Grant No.ZY20198003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: All authors know this paper and agree to submit it.

Acknowledgments: The authors would like to thank the referee for his or her careful reading and some comments on improving the presentation of this paper. Thanks Shengming Jiang of Shanghai Maritime University, Tomas Oppenheim and Zhuo Li, School of Engineering, University of California, for their comprehensive revision and some discussion of value.

Conflicts of Interest: We have no conflict of interest in the paper.

References

1. Teamah, A.A.M.; Afifi, W.A. Quasi-coordinate search for a randomly moving target. *J. Appl. Math. Phys.* **2019**, *8*, 1814–1825. [CrossRef]
2. Duvocelle, B.; Flesch, J.; Shi, H.; Vermeulen, D.s. Search for a moving target in a competitive environment. *Int. J. Game Theory* **2021**, *50*, 547–557. [CrossRef]
3. Uijlings, J.R.R.; van de Sande, K.E.A.; Gevers, T. Smeulders, Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
4. Silva, G.o.; Schnitman, L.; Oliveira, L. Constraining image object search by multi-scale spectral residue analysis. *Pattern Recognit. Lett.* **2014**, *39*, 31–38. [CrossRef]
5. Xin, Z.; Liao, G.; Yang, Z.; Huang, P.; Ma, J. A fast ground moving target focusing method based on first-order discrete polynomial-phase transform. *Digit. Signal Proc.* **2017**, *60*, 287–295. [CrossRef]
6. Hadidy, E.; Allah, M. Fuzzy optimal search plan for N-Dimensional randomly moving target. *Int. J. Comput. Method* **2016**, *13*, 38–38.
7. Shi, Z.; Wei, C.; Li, J.; Fu, P.; Jiang, S. Parallel search strategy in kernel feature space to track FLIR target. *Neurocomputing* **2016**, *214*, 671–683. [CrossRef]

8. Li, X.; Chen, J.; Deng, F.; Li, H. Profit-driven adaptive moving targets search with UAV swarms. *Sensors* **2019**, *7*, 1545. [CrossRef]
9. Song, T.; Huo, X.; Wu, X. A two-stage method for target searching in the path Planning for mobile robots. *Sensors* **2020**, *23*, 6919. [CrossRef]
10. Wong, W.K.; Ye, S.; Liu, H.; Wang, Y. Effective mobile target searching using robots. *Mob. Netw. Appl.* **2020**, 1–17. [CrossRef]
11. Pandey, K.K.; Parhi, D.R. Trajectory planning and the target search by the mobile robot in an environment using a behavior-based neural network approach. *Robotica* **2020**, *9*, 1627–1641. [CrossRef]
12. Khosraviani, M.; Zamani, M.S.; Bidkhori, G. FogLight: An efficient matrix-based approach to construct metabolic pathways by search space reduction. *Bioinformatics* **2016**, *32*, 398–408. [CrossRef]
13. Arya, K.V.; Gupta, P.; Kalra, P.K.; Mitra, P. Image registration using robust M-estimators. *Pattern Recognit. Lett.* **2007**, *28*, 1957–1968. [CrossRef]
14. Chiroma, H.; Herawan, T.; Fister, I.; Fister, I.; Abdulkareem, S.; Shuib, L.; Hamza, M.F.; Saadi, Y.; Abubakar, A. Bio-inspired computation: Recent development on the modifications of the cuckoo search algorithm. *Appl. Soft Comput.* **2017**, *61*, 149–173. [CrossRef]
15. Yang, X.; Deb, S. Cuckoo search: Recent advances and applications. *Neural Comput. Appl.* **2014**, *24*, 169–174. [CrossRef]
16. Walton, S.; Hassan, O.; Morgan, K. Selected engineering applications of gradient free optimisation using Cuckoo search and proper orthogonal decomposition. *Arch. Comput. Methods Eng.* **2013**, *20*, 123–154. [CrossRef]
17. Alsheddy, A. A two-phase local search algorithm for the ordered clustered travelling salesman problem. *Int. J. Metaheuristics* **2018**, *1*, 80–92. [CrossRef]
18. Bülentatay, G.E. Accelerating local search algorithms for the travelling salesman problem through the effective use of GPU. *Transp. Res. Procedia* **2017**, *22*, 409–418.
19. Yildiz, A.R. Cuckoo search algorithm for the selection of optimal machining parameters in milling operations. *Int. J. Adv. Manuf. Technol.* **2013**, *1–4*, 55–61. [CrossRef]
20. Ljouad, T.; Amine, A.; Rziza, M. A hybrid mobile object tracker based on the modified Cuckoo search algorithm and the Kalman filter. *Pattern Recognit.* **2014**, *11*, 3597–3613. [CrossRef]
21. Walia, G.S.; Kapoor, R. Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search. *Expert Syst. Appl.* **2014**, *41*, 6315–6326. [CrossRef]
22. de Menezes, D.Q.F.; Prata, D.M.; Secchi, A.R.; Pinto, J.C. A review on robust M-estimators for regression analysis. *Comput. Chem. Eng.* **2021**, *147*, 107254. [CrossRef]
23. Huillet, T.E. On Mittag-Leffler distributions and related stochastic processes. *J. Comput. Appl. Math.* **2016**, *296*, 181–211. [CrossRef]
24. Kwasnicka, H.; Przewozniczek, M. Multi population pattern searching algorithm: A new evolutionary method based on the idea of messy genetic algorithm. *IEEE Trans. Evol. Comput.* **2011**, *15*, 715–734. [CrossRef]

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
Tel. +41 61 683 77 34
Fax +41 61 302 89 18
www.mdpi.com

MDPI Books Editorial Office
E-mail: books@mdpi.com
www.mdpi.com/books

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland

Tel: +41 61 683 77 34

www.mdpi.com

ISBN 978-3-0365-5570-6